TITLE:

Robust Clustering Algorithm by Inner Product Scaling(Dissertation_全文)

AUTHOR(S):

Tsuda, Koji

CITATION:

Tsuda, Koji. Robust Clustering Algorithm by Inner Product Scaling. 京都大学, 1998, 博士(工学)

ISSUE DATE:

1998-03-23

URL:

https://doi.org/10.11501/3135511

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

Robust Clustering Algorithm by Inner Product Scaling

Koji Tsuda

1998

Abstract

The classification methods can be divided into the two types: supervised and unsupervised. In this thesis, we focus on the unsupervised classification (i.e. clustering). We will point out the properties that the clustering method should have, and propose a new clustering method with the properties.

The first property that the clustering method should have is the robustness against noise. The noises are defined as the isolated samples whose distances to the other samples are large. The robustness against noise is the property that the clustering result does not change when the noises are added to the sample set. The robustness against noise is important for the clustering method to be reliable even when the unexpected samples are included in the sample set.

The second property is the analytical computability. Usually, the clustering method is formulated as a nonlinear optimization problem and the clusters are obtained as the optimal solution. Since the optimization problem usually has many local minima, many initial values have to be tried, which takes a large computational time. When the clusters can be obtained by analytical computations without using the nonlinear optimization algorithm, the clustering method is said to be analytically computable. The analytical computability is important for keeping the computational time small.

In this thesis, we propose a clustering method which have both of the properties. We call this method "analytically-computable robust clustering method (ARC method)". The ARC method is derived as the extension of the extractive method, where the extension is performed by the mapping method called "inner product

scaling".

In the extractive method, the cluster center is determined so that the number of the samples in the neighborhood is maximized. When the cluster center is determined, the samples around the center are extracted as a cluster. Then, the extracted samples are removed from the sample set, and the next cluster is extracted in the same way. The cluster extraction stops when the predetermined number of clusters are extracted or the sample set becomes empty. One feature of the extractive method is the shape of the clusters can be changed by using various distance measures. For example, when the Euclidean distance is used, the shape of the cluster is spherical. When the cone distance is used, the clusters are cone-shaped.

The extractive method has the robustness against noise. On the other hand, in most choices of the distance measure, it does not have the analytical computability. However, it can be shown that, when the cone distance is used, the optimization problem of the extractive method can be solved analytically. But, the method has the drawback that it can only extract cone-shaped clusters, although the actual clusters are considered to be spherical in many cases. So, the mapping method that converts the spherical clusters into the cone-shaped clusters is needed for the preprocessing.

For this purpose, the mapping method called "inner product scaling" is used. By this method, the samples in the feature space are mapped into another space so that the Gaussian similarity between the samples is reflected as the inner product. We will show that the inner product scaling can convert the spherical clusters into the cone-shaped clusters. The ARC method is the combination of the inner product scaling and the cone cluster extraction. Since the two elements are analytically computable, the ARC method can perform the spherical cluster extraction by analytical computations.

In regard to the robustness against noise, we compared the ARC method with the Noise Resistent C-Means method, which is a partitional clustering method specially modified to improve the robustness against noise. As a result, the ARC method

achieved higher robustness.

We will present three applications of the ARC method. First, the ARC method is applied to image processing: extraction of lines from an image. Clustering line segments is an effective approach for line extraction. Line segments are obtained by applying a line fitting process to the output of edge detection process. The similarity between the line segments is defined so that two segments aligned in line have a large similarity. Then, lines are extracted as clusters of line segments. The line segments which are not aligned in line are considered as noises. The noise robustness of the ARC method works well in the line extraction task.

Second, the ARC method is applied to document clustering. We used document clustering for browsing a document database. The outline of the clustering-based browsing system is as follows: The documents are indexed by the term occurrence frequency and the similarity between documents is defined based on the number of common terms. Similar documents are clustered and a representative document of each cluster is shown to the user. The user can get the whole view of the database without examining documents one by one. In the document database, there are many documents whose contents are not similar to any document. Such documents are considered as noise documents in clustering, so the ARC method is also useful in document clustering.

Third, the ARC method is applied to the prototype generation for the nearest neighbor method. To reduce the computational cost of the nearest neighbor method, the reduction of the training samples is required. The training samples are partitioned into several clusters and the reduced training set (i.e. the prototype set) is obtained as the cluster centers. It is known that a "noise" training sample which is distant from the other training samples is harmful for classifiers. So, the robust clustering is needed to generate the prototypes that achieve high classification accuracy. We will show that the ARC method can generate better prototypes than the C-Means with respect to the classification accuracy of the nearest neighbor method.

Contents

List of Symbols

p	Dimensionality of the feature space
$\Re^p$	p-dimensional real Euclidean space (the feature space)
$\mathbf{t}_i$	Sample in $\Re^p$
$d(\mathbf{t}_i, \mathbf{t}_j)$	Distance between the two samples $\mathbf{t}_i$ and $\mathbf{t}_j$.
$\mathcal{T}$	Set of samples
n	Number of samples in $\mathcal{T}$
$\mathcal{C}_k$	Cluster of samples, which is a subset of $\mathcal{T}$
c	Number of clusters
n_k	Number of samples in $\mathcal{C}_k$
u_{ki}	Membership value of $\mathbf{t}_i$ to $\mathcal{C}_k$
$\mathbf{u}_k$	n-dimensional membership vector of $\mathcal{C}_k$
$\mathbf{m}_k$	Cluster center of $\mathcal{C}_k$ in $\Re^p$
$\Re^n$	n-dimensional real Euclidean space (the image space of the inner product scaling)
$\boldsymbol{\tau}_i$	Sample in $\Re^n$ mapped by the inner product scaling
$\boldsymbol{\mu}_k$	Cluster center of $\mathcal{C}_k$ in $\Re^n$
ν	Width parameter of the window function
σ	Parameter to control the cluster size
η	Cluster boundary parameter in the ARC method
δ	Distance threshold in the NR C-Means method

Chapter 1

Introduction

Classification is the task to assign an object to one of the classes. There are two cases in classification: unsupervised and supervised. In the unsupervised classification, the classes are not given a priori but are created by gathering similar objects. The unsupervised classification is also called "clustering"[1]. The clustering algorithms play a fundamental role in many fields of computer science such as artifificial intelligence[2, 3], communication[4] and information retrieval[5], because the organization of data into clusters is one of the most important fundamental procedures of understanding and learning. On the other hand, in the supervised classification, the definition of each class is given a priori. The classifying rule is created from the class definition, and a sample is classified to the class based on the rule. Supervised classification is also called "pattern recognition"[6].

To implement classification algorithms on computers, the samples must be represented by numerical data. In common settings[6], a sample is represented by a tuple of p measurements. The sample is described as

$$\mathbf{t} = (t_1, \cdots, t_p)^T, \tag{1.1}$$

where $t_j (j = 1, \cdots, p)$ is the real value which corresponds to the j-th measurement, and ^{T} denotes the transpose of a vector or a matrix. Since $\mathbf{t}$ is a p-dimensional real vector, the sample is considered as a point in the p-dimensional real space $\Re^p$. This

space is called a "feature space".

The purpose of clustering is to extract c subsets (i.e. clusters) $\mathcal{C}_1, \cdots, \mathcal{C}_c$ out of the finite sample set

$$\mathcal{T} = \{\mathbf{t}_1, \cdots, \mathbf{t}_n\}, \tag{1.2}$$

where each subset is comprised of *similar* samples. The clustering is performed based on "distance" $d(\mathbf{x}, \mathbf{y})$, which is the function defined on $\Re^p \times \Re^p$. The distance shows how two samples are not similar. When the distance is large, the two samples are considered as not similar. The distance has the following properties [1] :it is symmetric,

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}), \tag{1.3}$$

nonnegative,

$$d(\mathbf{x}, \mathbf{y}) \geq 0, \tag{1.4}$$

and becomes zero when $\mathbf{x} = \mathbf{y}$,

$$d(\mathbf{x}, \mathbf{x}) = 0. \tag{1.5}$$

The most frequently used distance is the Euclidean distance denoted as follows:

$$\| \mathbf{x} - \mathbf{y} \| = \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2}. \tag{1.6}$$

Clustering is used in the fields of computer science for many tasks. The tasks can be classified into two categories.

- Compression Tasks

  In these tasks, the samples in a cluster are replaced by a representative sample. Then, the number of samples is reduced to the number of clusters, and so the storage for the samples is compressed. The clustering methods are evaluated by the compression rate and the compression quality, where the compression rate describes how much the storage reduced, and the compression quality is defined by the average distance between each sample in the cluster and the representative sample. For example, the vector quantization[4] is a typical data compression task.

- Extraction Tasks

  In these tasks, a cluster is expected to correspond to an entity in a real world. The clustering methods are evaluated by the correspondence between a cluster and a real entity. For example, in the image segmentation task[8], an image is segmented to a number of regions so that a region corresponds to an object in the image. This task can be implemented as the clustering of pixels, where the distance between two pixels is defined as the probability that they are contained in different regions. A cluster of pixels is expected to correspond to a region in the image.

[1] Notice that the triangle equality is not included here. The distance which also satisfies the triangle inequality is called a "metric"[7].

In the artificial intelligence or the pattern recognition, the extraction tasks are of main interest[6]. For the extraction tasks, the clustering method should have the following two properties, that is, *the robustness against noise* and *analytical computability*.

The reason why the robustness against noise is needed is explained as follows. In extraction tasks, the presence of *noises* makes clustering difficult. Conceptually, the noises are defined as the samples that do not comprise the entity that we want. In the task of image segmentation, the isolated pixels that do not belong to any region are considered as noises. Quantitatively, the noise is defined as the isolated sample in the feature space whose distances to the others are large (Fig. 1.1). Also, the robustness against the noise is defined as the property that the clustering result does not change by the addition of the noises. When the image segmentation is performed by the robust methods[9], the produced regions do not change when the isolated pixels are added. But, in the non-robust methods such as the K-Means

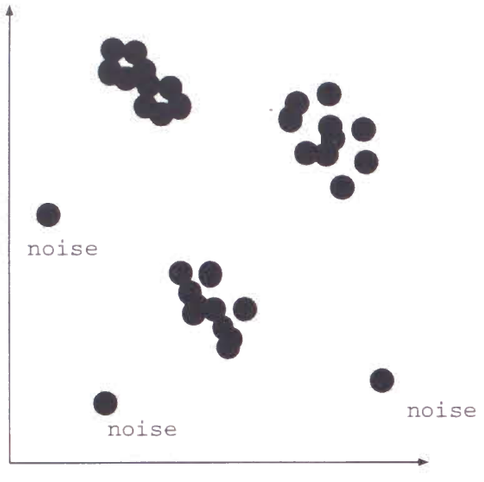


Figure 1.1: Noises in clustering

method[1], the produced regions are biased by the isolated pixels and the regions are not correctly extracted.

We will also explain the concept of the analytical computability. In the extraction tasks, the number of samples could be large (e.g. over 1000) and the number of clusters could also be large (e.g. over 100). So, it is important that the computational time of the clustering method is admissible even in such large scale problems. The analytical computability works for keeping the computational time small.

Most clustering methods are formulated as the nonlinear optimization problem[1]: find the parameter vector $\boldsymbol{\theta} \in \Re^q$ that minimize the objective function $f(\boldsymbol{\theta}) \in \Re$. To solve the nonlinear optimization problem, you have to adopt the gradient descent strategy as follows: First, a random initial value is set to $\boldsymbol{\theta}$. Then, the gradient vector of $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}$ is computed and $\boldsymbol{\theta}$ is changed slightly to the reverse direction of the gradient. By repeating this process, finally get to the point in the parameter space of $\boldsymbol{\theta}$ where the gradient becomes the zero vector (i.e. saddle point). Then, the process ends at this point. But, when the objective function is nonlinear, this point is not necessarily the minimum of the objective function. In such a case, this point

is called a "local minimum". To seek the global optimal solution, you have to try many initial points to avoid local minima, which takes large computational time. By the word "analytically-computable clustering method", we mean the clustering method where the clusters can be obtained by analytical computations without using the gradient descent method. In such methods, the local minima problem does not occur and the computational time is usually smaller than the one which is not analytically computable.

In recent years, it is widely recognized that the robustness against noise is an important property for clustering[10]. Several robust clustering methods have been proposed[11, 12, 13], but none of these methods are analytically computable. On the other hand, the importance of the analytical computability is also recognized, but it is considered to be difficult to implement the clustering algorithms by the analytical computations. In recent years, there are almost no studies to propose the analytically computable clustering method.

In this thesis, we will propose the clustering method that has the two properties. We call this method "analytically-computable robust clustering method (ARC method)". This method is derived as the extension of the clustering method called "the extractive method". In the extension, the key point is the mapping method, which is called the "inner product scaling".

In the extractive method, the cluster center $\mathbf{m} \in \Re^p$ is determined so that the number of the samples in the neighborhood of $\mathbf{m}$ is maximized. When the cluster center is determined, the samples around $\mathbf{m}$ are extracted as a cluster. The extracted samples are removed from the sample set, and the next cluster is extracted in the same way. The cluster extraction stops when the predetermined number of clusters are extracted or the sample set becomes empty.

The mathematical formulation of the extractive method is described as follows: Let the set $\mathcal{N}(\mathbf{m}) \subset \Re^p$ be the neighborhood of $\mathbf{m}$, then it is described as

$$\mathcal{N}(\mathbf{m}) = \{\mathbf{x} | d(\mathbf{x}, \mathbf{m}) \leq \rho\}, \tag{1.7}$$

where $\rho \in \Re$ ($\rho > 0$) is the width of the neighborhood. The number of samples in

the neighborhood is written as

$$p(\mathbf{m}) = \sum_{i=1}^{n} w(d(\mathbf{t}_i, \mathbf{m})). \tag{1.8}$$

where $w(x)$ is a *window function* described as,

$$w(x) = \begin{cases} 1 & (x \leq \rho) \\ 0 & (x > \rho) \end{cases}. \tag{1.9}$$

The cluster center is seeked to maximize the number of samples. This optimization problem is formulated as follows:

$$\text{Find } \mathbf{m} \text{ that maximizes p}(\mathbf{m}). \tag{1.10}$$

When the optimal center is found, the samples in the neighborhood $\mathcal{N}$ are extracted as a cluster. But, when the window function in (1.9) is used, $p(\mathbf{m})$ is not continuous and so the optimization is very difficult. So, the smoothed window functions are used in many cases. The most frequently used one is the Gaussian window function:

$$w_g(x) = \exp(-\frac{x^2}{\sigma^2}), \tag{1.11}$$

where σ is the parameter which determines the width of the window function. In the case of the smooth window functions, the samples whose $w_g(d(\mathbf{t}_i, \mathbf{m}))$ is above the threshold are extracted as a cluster.

The main strong point of the extractive clustering method is the robustness against noise. We will show that the extractive clustering method is robust against noise by an example. Fig. 1.2(a) shows the situation that the samples are distributed in the two dimensional space and one cluster is extracted by the extractive method. Here, we assume that the distance is defined as the Euclidean distance and the window function in (1.9) is used. The dotted circle in the figure denotes the boundary of the cluster. On the other hand, Fig. 1.2(b) shows the situation where a number of noises are added to the sample set. When the cluster center is set in the areas of noises, the number of samples in the neighborhood is small, because the noises are apart from each other. So, the cluster center is not to be determined in the areas of noises. As a result, the cluster center is not moved from the one in the case without

noises. Since the noises do not affect the clustering result, the extractive method is considered as robust against noise.

On the other hand, the extractive clustering method is not analytically computable for most choices of the distance measure and the window function. But, when particular distance measure and window function are chosen, the extractive clustering method becomes analytically computable. Let the distance be

$$d_c(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^T \mathbf{y}}{\| \mathbf{x} \| \| \mathbf{y} \|} \tag{1.12}$$

and the window function be

$$w_c(x) = (1 - \frac{x}{\nu})^2, \tag{1.13}$$

where ν is the width parameter. Then, the optimization problem of (1.10) is rewritten as follows: Find $\mathbf{m}$ that maximizes

$$p_c(\mathbf{m}) = \sum_{i=1}^{n} (\frac{\nu - 1}{\nu} + \frac{\mathbf{m}^T \mathbf{t}_i}{\nu \| \mathbf{m} \| \| \mathbf{t}_i \|})^2. \tag{1.14}$$

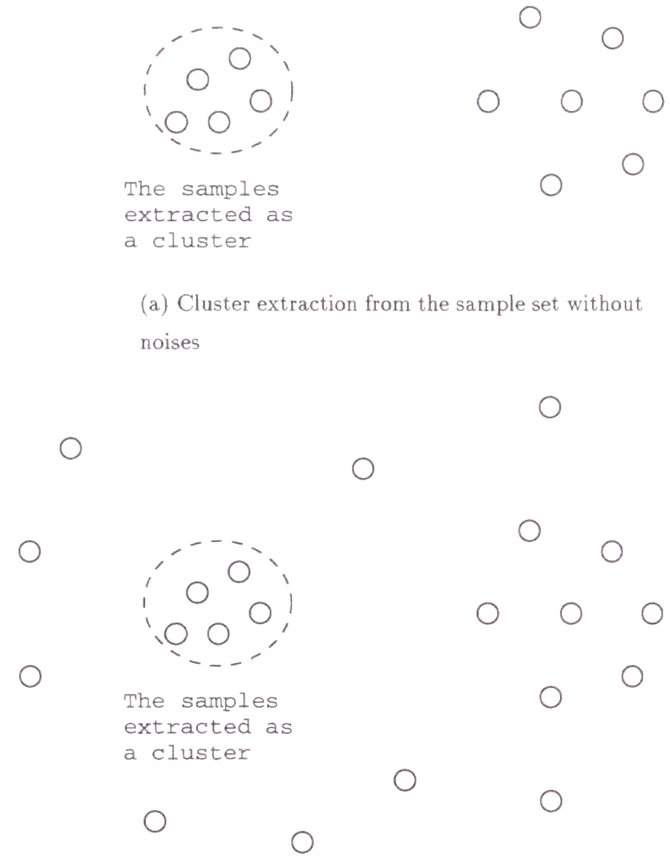
When we further assume that $\nu = 1$, the optimization problem is rewritten as: Find $\mathbf{m}$ that maximizes

$$p_c(\mathbf{m}) = \sum_{i=1}^{n} (\frac{\mathbf{m}^T \mathbf{t}_i}{\| \mathbf{m} \| \| \mathbf{t}_i \|})^2. \tag{1.15}$$

This optimization problem is identical with the one used in the principal component analysis[6], and the optimal solution can be obtained by the analytical computations (The proof will be shown in Sec. 3.3).

The contour plot of the distance $d_c(\mathbf{x}, \mathbf{y})$ in the 2-dimensional space is shown in Fig. 1.3, where $\mathbf{y}$ is fixed to $(1, 1)$ and $\mathbf{x}$ is moved. Since the contour of the distance $d_c(\mathbf{x}, \mathbf{y})$ forms the cone whose central axis is the line that intersects the origin and $\mathbf{y}$, we call the distance "the cone distance". When using the cone distance, the extracted clusters are also cone-shaped. From later on, we call the extractive method with the cone distance "the cone cluster extraction".

In the cone cluster extraction, we have the analytical computability. But, there is a critical drawback in the cone cluster extraction: it only extracts the cone-shaped



(a) Cluster extraction from the sample set without noises

(b) Cluster extraction from the sample set with noises

Figure 1.2: When a number of noises are added to the samples, the cluster extracted by the extractive method remains the same.

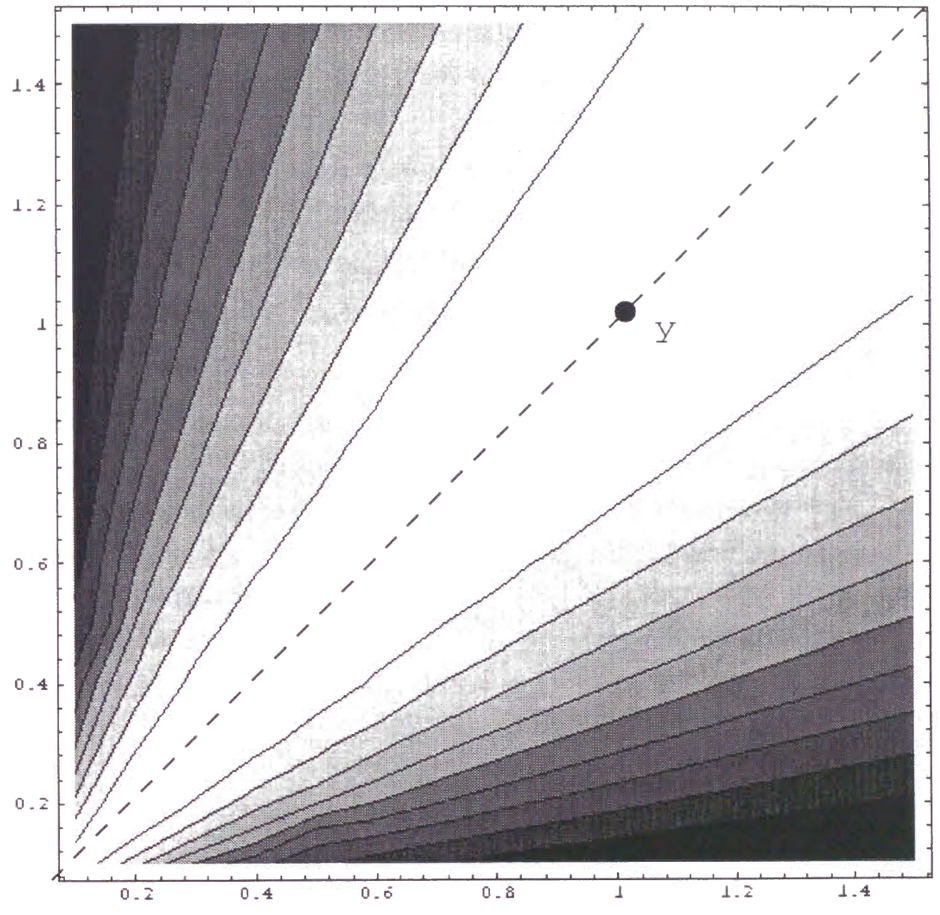


Figure 1.3: The contour plot of the cone distance. As the color changes from white to black, the distance becomes larger.

cluster of the fixed size. Practically, the shape and the size of the clusters are various according to the tasks. Therefore, the applicable tasks of the cone cluster extraction are very limited.

In most clustering methods, the shape of clusters is assumed to be spherical[1], and such clustering methods have been successfully applied to many tasks. To make the cone cluster extraction applicable to many tasks, the mapping of the samples is needed to convert the spherical clusters into the cone-shaped clusters. Also, the mapping should be able to control the size of cone-shaped clusters so that they can be extracted correctly by the cone cluster extraction.

We propose the mapping method called "inner product scaling"[14] for this purpose. This method maps the samples $\mathbf{t}_1, \cdots, \mathbf{t}_n$ to the ones $\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_n$ in n-dimensional space $\Re^n$:

$$\mathbf{t}_i \in \Re^p \rightarrow \boldsymbol{\tau}_i \in \Re^n, \ (i = 1, \cdots, n). \tag{1.16}$$

The purpose of this method is to map the samples so that the similarity defined as the Gaussian function (i.e. Gaussian similarity),

$$s(\mathbf{t}_i, \mathbf{t}_j) = \exp(-\frac{\| \mathbf{t}_i - \mathbf{t}_j \|^2}{\sigma^2}) \tag{1.17}$$

is reflected as the inner product $\boldsymbol{\tau}_i^T \boldsymbol{\tau}_j$. The mapped samples $\boldsymbol{\tau}_i (i = 1, \cdots, n)$ are obtained so that the following simultanous equations are satisfied:

$$s(\mathbf{t}_i, \mathbf{t}_j) = \boldsymbol{\tau}_i^T \boldsymbol{\tau}_j, \ (i, j = 1, \cdots, n). \tag{1.18}$$

The proof that the mapped samples can be obtained by analytical computations will be shown in Sec. 3.2.

By the inner product scaling, the samples of the spherical cluster in $\Re^p$ are mapped to the cone cluster in $\Re^n$. Assume the cluster $\mathcal{C}_k$ has the n_k samples $\mathbf{t}_{k1}, \cdots, \mathbf{t}_{kn_k}$, and the following inequality holds:

$$\| \mathbf{t}_{ki} - \mathbf{t}_{kj} \|^2 \leq r, \ (i = 1, \cdots, n_k), \tag{1.19}$$

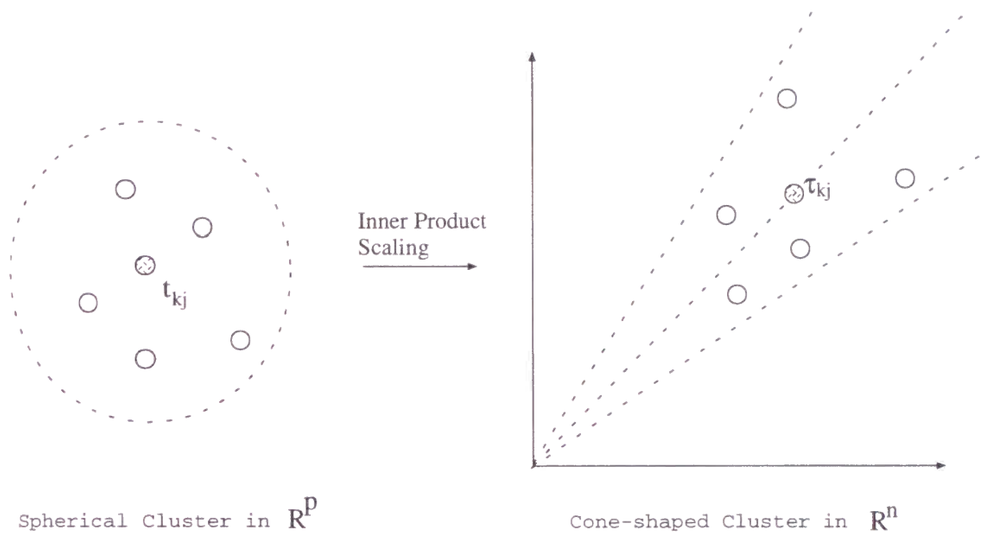


Figure 1.4: The inner product scaling maps a spherical cluster into a cone-shaped cluster

where $\mathbf{t}_{kj}$ is the sample in $\mathcal{C}_k$ which is nearest to the cluster center. This inequality shows that the cluster is spherical, i.e., the samples of the cluster are inside the sphere of radius r centered on $\mathbf{t}_{kj}$. When the samples are mapped on $\Re^n$ by the inner product scaling, the mapped samples satisfy the following inequality:

$$\frac{\boldsymbol{\tau}_{ki}^T \boldsymbol{\tau}_{kj}}{\| \boldsymbol{\tau}_{ki} \| \| \boldsymbol{\tau}_{kj} \|} \geq \gamma(r, \sigma), \ (i = 1, \cdots, n_k), \tag{1.20}$$

where

$$\gamma(r, \sigma) = \exp(-\frac{r}{\sigma^2}). \tag{1.21}$$

Therefore, the spherical cluster C_k in $\Re^p$ are mapped to the cone-shaped cluster whose central point is $\boldsymbol{\tau}_{kj}$ and radius is $\gamma(r, \sigma)$, which is illustrated in Fig. 1.4. Also, the size of the cone-shaped cluster is described by $\gamma(r)$ and can be controlled by the parameter σ.

The ARC method is the combination of the inner product scaling and the cone cluster extraction. Since the two elements are analytically computable, the ARC method can perform the spherical cluster extraction by analytical computations. So far, the extractive method have not been applied to the actual large-scale problems, although it has the robustness against noise. It is because of its large computational time caused by the repeated trials to avoid local minima. Now, the ARC method overcomes the drawback, so it can be applied to the actual problems.

In this thesis, we will show three applications of the ARC method. First, the ARC method is applied to the image processing: the extraction of lines from an image. Clustering line segments is known as an effective approach for line extraction. Line segments are obtained by applying the line fitting process to the output of the edge detection process. The similarity is defined between the pairs of line segments so that the two segments aligned in line have a large similarity. Then, lines are extracted as the clusters of line segments. The line segments which are badly aligned in line are considered as noises. So, the noise robustness of the ARC method works well in the line extraction task.

Second, the ARC method is applied to the document clustering. We used the document clustering for browsing a document database[15]. The outline of the clustering-based browsing system is as follows: The documents are indexed by the term occurrence frequency and the similarity between pairs of documents is defined based on the number of common terms. The documents which have the large similarity to each other are clustered and a *representative document* of each cluster is shown to the user. The user can get the whole view of the database without examining the documents one by one. In the document database, there are many documents whose contents are not similar to any document. Since such documents are considered as noise documents, the ARC method is useful also in the document clustering.

Third, the ARC method is applied to the prototype generation for pattern recognition. The nearest neighbor pattern recognition method[6] requires the distance to

every training sample for classification. Since the computational time to obtain every distance is very high, the reduction of the training samples is required to reduce the computational time. Clustering methods are often used for this purpose. The training samples are partitioned into several clusters and the reduced training set (i.e. the prototype set) is obtained as the cluster centers. The most frequently used method for the prototype generation is the C-Means method[1, 16], but it is easily affected by the noise training sample which is the training sample isolated from the others. We will show that the ARC method can generate the prototypes which achieve high classification accuracy because of the robustness against noise. To compare the prototypes generated by the C-Means and the ARC method, we performed two pattern recognition experiments. The first one is the appearance-based 3D object recognition experiment[17], and the second one is the Hiragana recognition experiment[18, 19]. As a result, the prototypes generated by the ARC method achieved higher classification accuracy.

The rest of this thesis is organized as follows: In Chap. 2, the previous clustering methods are briefly reviewed. In Chap. 3, the ARC method is described in detail. We describe the two factors that makes this method: the inner product scaling and the cone clustering extraction. Also, the robustness against noise of this method is evaluated in comparison with NR C-Means[11] method. Chap. 4, 5 and 6 will present the three applications: line extraction, document clustering and prototype generation, respectively. Chap. 7 is the concluding remarks.

Chapter 2

Prior Clustering Methods

This chapter presents prior clustering methods, which will be compared with the ARC method in the later chapters.

There are three types in the clustering method: extractive, partitional and agglomerative. The main difference between the types is the assumption which the clustering method imposes on the sample set. The three types are summarized as follows:

- Extractive methods

  The size of clusters is assumed in advance.

- Partitional methods

  The number of clusters is assumed in advance.

- Agglomerative methods

  No assumptions are made for the sample set. The number and the size of clusters can be determined after clustering.

Before proceeding to the description of the clustering methods, we introduce the *membership value notation* of the clusters. The membership value u_{ki} is often used to represent the clusters, when it is not convenient to describe a cluster as a set[1]. The membership value u_{ki} stands for the degree that the sample $\mathbf{t}_i$ belongs to the

cluster $\mathcal{C}_k$. When the membership values are limited to either 0 or 1, each sample belongs to the only one cluster. Such a clustering method is called a *hard clustering method*. On the other hand, when the membership values are allowed to be any real values between 0 and 1, the sample is allowed to belong to several clusters at the same time. Such a clustering method is called a *fuzzy clustering method*.

2.1 Extractive Methods

In the extractive methods, the cluster center $\mathbf{m} \in \Re^p$ is determined so that the number of the samples in the neighborhood of $\mathbf{m}$ is maximized. The size of neighborhood is determined in advance, which will be the size of the cluster. When the cluster center is determined, the samples in the neighborhood are extracted as a cluster. The extracted samples are removed from the sample set, and the next cluster is extracted in the same way. The cluster extraction stops when the predetermined number of clusters are extracted or the sample set becomes empty.

The optimization problem for seeking the center is formulated as follows:

$$\text{Find } \mathbf{m} \text{ that maximizes p}(\mathbf{m}), \tag{2.1}$$

where the objective function $p(\mathbf{m})$ is described as

$$p(\mathbf{m}) = \sum_{i=1}^{n} w(d(\mathbf{t}_i, \mathbf{m})). \tag{2.2}$$

Here, $w(x)$ is the window function which characterizes the extent of the neighborhood of the center. The most frequently used window function is the Gaussian window function described as follows:

$$w_g(x) = \exp(-\frac{x^2}{\sigma^2}), \tag{2.3}$$

where the size of the neighborhood is controlled by the parameter σ. When the optimal center is determined, the samples whose $w_g(d(\mathbf{t}_i, \mathbf{m}))$ is above the threshold are extracted as a cluster.

The strong point of the extractive method is the robustness against noise[13], and the weak point is that the optimization problem has many local minima[13]. The examples of the extractive methods include the mode seeking method[1, 20] and Jolion's method[13].

2.2 Partitional Methods

2.2.1 K-Means Method

The K-Means method is a hard clustering method where the number of cluster c is determined in advance, and the sample set $\mathcal{T}$ is divided into c disjoint clusters $\mathcal{C}_1, \cdots, \mathcal{C}_c$. In the membership value notation, the membership value u_{ki} is limited to 0 or 1,

$$u_{ki} = \{0, 1\}, \tag{2.4}$$

and the membership values indicate 1 only for the cluster,

$$\sum_{k=1}^{c} u_{ki} = 1. \tag{2.5}$$

The K-Means method is formulated as the optimization problem that minimizes the error function that describes the badness of the clusters. Since the purpose of clustering is to extract mutually similar samples, the badness of the cluster $\mathcal{C}_k$ is measured by how far the samples are apart from the cluster center. So, the error function of each cluster $\mathcal{C}_k$ is described as the sum of the squared distance from the cluster center $\mathbf{v}_k$ as follows:

$$\sum_{i=1}^{n} u_{ki} d^2(\mathbf{v}_k, \mathbf{x}), \tag{2.6}$$

where the cluster center is described as

$$\mathbf{v}_k = \sum_{i=1}^{n} u_{ki}\mathbf{x} / \sum_{i=1}^{n} u_{ki}. \tag{2.7}$$

The error function of all clusters is described as the sum of that of each cluster:

$$g(\mathbf{u}_1, \cdots, \mathbf{u}_c) = \sum_{k=1}^{c} \sum_{i=1}^{n} u_{ki} d^2(\mathbf{v}_k, \mathbf{x}), \tag{2.8}$$

where $\mathbf{u}_k$ denotes the n-dimensional vector whose i-th element is u_{ki}. The K-Means method is formulated as the optimization problem as follows: Find $\mathbf{u}_k (k = 1, \cdots, c)$ that minimizes the error function $g(\mathbf{u}_1, \cdots, \mathbf{u}_c)$ subject to the constraints of (2.4) and (2.5).

Such a nonlinear optimization problem is usually solved by the gradient descent algorithm. But, for K-Means, there is a specialized optimization algorithm[1]. In this algorithm, the optimization can be performed by repeating two phases alternately. The first phase sets the centers at the means of the samples in clusters. The second phase assigns every sample to its nearest center and forms clusters. The detailed algorithm is described as follows:

1. Initialization.

   Set the initial clusters $\mathcal{C}_k (k = 1, \cdots, c)$ randomly or using a heuristic technique[21].

   Let $g_{old} \leftarrow \infty$, where $\leftarrow$ denotes the substitution.

2. Obtain the cluster centers.

   Calculate the cluster centers $\mathbf{v}_k$ of all clusters as (2.7).

3. Update the clusters.

   For each sample $\mathbf{t}_i (i = 1, \cdots, n)$, the nearest cluster $\mathcal{C}_\ell$ is determined such that the cluster center $\mathbf{v}_\ell$ is the nearest among the cluster centers. Then, the sample is added to $\mathcal{C}_\ell$. In the membership value notation, this process can be written as follows:

   $$u_{\ell i} \leftarrow 1 \tag{2.9}$$

   $$u_{ki} \leftarrow 0 \ (k \neq \ell) \tag{2.10}$$

   where ℓ is the index of the class which has the nearest center:

   $$\ell = \arg \min_{k=1,\cdots,c} d(\mathbf{v}_k, \mathbf{t}_i). \tag{2.11}$$

   As a result, the sample set $\mathcal{T}$ is divided thoroughly into the disjoint clusters.

4. Calculate the error function $g(\mathbf{u}_1, \cdots, \mathbf{u}_c)$ as (2.8).

5. Convergence check.

   Stop if $|g(\mathbf{u}_1, \cdots, \mathbf{u}_c) - g_{old}| < \epsilon$, where ϵ is stopping criterion. Otherwise, $g_{old} \leftarrow g(\mathbf{u}_1, \cdots, \mathbf{u}_n)$ and go to step 2.

It is proven that, by this algorithm, a local minimum of the error function can be found[1]. But, it is not guaranteed to reach to the global minimum. To seek the good solution comparable to the global minimum, many trials with changing the initial clusters are required. So, the practical computational time for the K-Means method is substantially large.

2.2.2 C-Means Method

The C-Means method is the fuzzy version of the K-Means method[16]. The difference is that the membership value is allowed to be any value between 0 and 1. The error function of the C-Means method is given as follows:

$$g(\mathbf{u}_1, \cdots, \mathbf{u}_c) = \sum_{k=1}^{c} \sum_{i=1}^{n} u_{ki}^{\mu} d(\mathbf{t}_i, \mathbf{v}_k)^2, \tag{2.12}$$

and the constraints for the membership values are described as follows:

$$\sum_{k=1}^{c} u_{ki} = 1, \quad 0 \leq u_{ki} \leq 1. \tag{2.13}$$

The difference between the two error functions is that the "fuzzification parameter" μ $(1 \leq \mu \leq \infty)$ is added in the C-Means method. When μ equals to 1, each membership value u_{ki} converges to either 0 or 1 in minimizing the error function g[16]. For the clusters to be fuzzy, μ should be determined to the value more than 1. But, when μ is too large, all the membership values converge to the same value. It is known that the practical range of μ is $1 \leq \mu \leq 5$[8].

The optimization algorithm of the K-Means is applicable to the C-Means algorithm with the following two modifications: In Step. 2, the cluster center is calculated as follows:

$$\mathbf{v}_k = \sum_{i=1}^{n} u_{ki}^{\mu} \mathbf{t}_i / \sum_{i=1}^{n} u_{ki}^{\mu}. \tag{2.14}$$

In Step. 3, the membership values are set as follows:

$$u_{ki} = (\sum_{\ell=1}^{c} d(\mathbf{v}_k, \mathbf{t}_i)^2 / d(\mathbf{v}_\ell, \mathbf{t}_i)^2)^{-1/(\mu-1)}. \tag{2.15}$$

2.2.3 Noise Resistent C-Means

In Noise Resistent C-Means (NR C-Means)[11], the noise cluster is added as the $(c+1)$-th cluster to achieve robustness against noise. This algorithm is designed to classify noises to the noise cluster. The algorithm is obtained by modifying the step 3 of C-Means as follows:

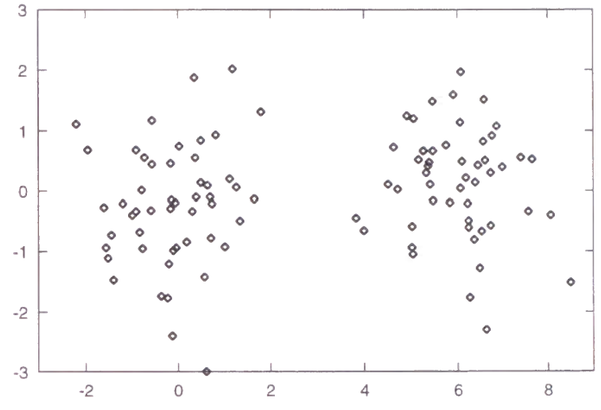
$$u_{ki} = \begin{cases} (\sum_{\ell=1}^{c} d(\mathbf{v}_k, \mathbf{t}_i)^2 / d(\mathbf{v}_\ell, \mathbf{t}_i)^2)^{-1/(\mu-1)} & (1 \le k \le c) \\ \delta & (k = c+1) \end{cases}, \tag{2.16}$$

where $\delta \in \Re$ $(\delta > 0)$ is the distance threshold. As a result, the samples whose distance to the nearest cluster center is more than δ are gathered to the noise cluster. So, the isolated noises are assigned to the noise cluster. Several algorithms based on similar principle are reported by Ichikawa[22] and Frigui[12].

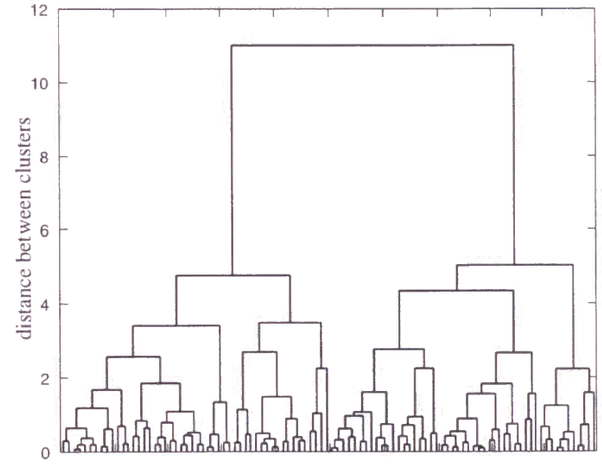
2.3 Agglomerative Methods

In the agglomerative methods, each sample forms a cluster at the beginning, and it is repeated that the two clusters which has the smallest distance are merged to a single cluster. The result of merging clusters is visualized in a *dendrogram*(Fig.2.1). By cutting the dendrogram at a certain level, you can obtain any number of clusters. So, in the agglomerative methods, you need not to specify the number of clusters in advance.

The agglomerative methods vary with the definition of the distance between two clusters. The representative methods are Single Link[23], Complete Link and Group Average[1]. In Single Link, the distance between the clusters is defined as the smallest distance between the samples in the clusters as follows:



(a) 100 samples in the 2-dimensional space



(b) Dendrogram

Figure 2.1: The dendrogram is derived from the 100 samples in the 2-dimensional space using Complete Link. The distance between the samples was defined by their Euclidean distance. The vertical axis of the dendrogram denotes the distance between the merged clusters. You can see two distinct clusters in the dendrogram.

$$d(\mathcal{C}_1, \mathcal{C}_2) = \min_{\mathbf{t}_i \in \mathcal{C}_1, \mathbf{t}_j \in \mathcal{C}_2} d(\mathbf{t}_i, \mathbf{t}_j). \tag{2.17}$$

On the contrary, in Complete Link, the distance between the clusters is defined as the largest distance between the samples.

$$d(\mathcal{C}_1, \mathcal{C}_2) = \max_{\mathbf{t}_i \in \mathcal{C}_1, \mathbf{t}_j \in \mathcal{C}_2} d(\mathbf{t}_i, \mathbf{t}_j). \tag{2.18}$$

In Group Average, the distance between clusters is defined as the average of the distances between the samples.

$$d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{n_1 n_2} \sum_{\mathbf{t}_i \in \mathcal{C}_1} \sum_{\mathbf{t}_j \in \mathcal{C}_2} d(\mathbf{t}_i, \mathbf{t}_j), \tag{2.19}$$

where n_1 and n_2 are the numbers of the samples in $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

For Single Link, there is a fast $O(n^2)$ algorithm[24], but the resultant clusters are very "loose", that is, the distances between two samples within a cluster is not remarkably smaller than the distances between the samples of different clusters. Therefore, Single Link is seldom used in literature[24] except for rough and preliminary exploration of data. On the contrary, Complete Link makes "tight" clusters which are useful for many applications. But it requires $O(n^3)$ time[1], which is not realistic for a large set.

Chapter 3

Robust Clustering by Inner Product Scaling

3.1 Introduction

In this chapter, we describe the analytically-computable robust clustering method (i.e. the ARC method). The ARC method is the clustering method such that the clusters can be obtained by the analytical computations. Also, it has the robustness against noise, which is useful for the tasks where many noises are expected to be included in the sample set.

The ARC method is derived as the extension of the cone cluster extraction method, which is the extractive clustering method based on the cone distance. The cone cluster extraction has the advantage that the clusters can be obtained by the analytical computations, but also has the drawback that it can only extract the cone-shaped clusters of the fixed size. Since the actual clusters are considered to be spherical, we added the preprocessing called "inner product scaling" which converts the spherical clusters into the cone-shaped clusters.

In this chapter, we will describe the two elements of the ARC method: the inner product scaling and the cone cluster extraction. We will show the proofs that the two elements can be performed by analytical computations. Also, the robustness

against noise of the ARC method is evaluated by comparing it with the Noise Resistent C-Means method proposed by Dave[11]. The two methods are applied to an artificial point set including spherical clusters and uniformly distributed noises. We examined the number of clusters correctly extracted. As a result, the ARC method achieved higher robustness, which shows the effectiveness of the ARC method in noisy problems.

The rest of this chapter is organized as follows: In Sec. 3.2, the inner product scaling is explained. In Sec. 3.3, the cone cluster extraction is described. In Sec. 3.4, the robustness of the ARC method is compared with that of NR C-Means method. Sec. 3.5 is the summary.

3.2 Inner Product Scaling

As the first phase of the ARC method, the inner product scaling is performed to convert the spherical clusters into the cone-shaped clusters. The inner product scaling maps the samples $\mathbf{t}_1, \cdots, \mathbf{t}_n \in \Re^p$ to the samples $\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_n \in \Re^n$, respectively. The mapping is performed so that the Gaussian similarity in $\Re^p$

$$s(\mathbf{t}_i, \mathbf{t}_j) = \exp(-\frac{\| \mathbf{t}_i - \mathbf{t}_j \|^2}{\sigma^2}) \tag{3.1}$$

is reflected as the inner product $\boldsymbol{\tau}_i^T \boldsymbol{\tau}_j$, where σ is a scalar parameter. This condition is described as the following simultanous equations:

$$s(\mathbf{t}_i, \mathbf{t}_j) = \boldsymbol{\tau}_i^T \boldsymbol{\tau}_j, \quad (i, j = 1, \cdots, n). \tag{3.2}$$

We are going to explain how to solve the simultanous equations (3.2) to obtain $\boldsymbol{\tau}_i (i = 1, \cdots, n)$. In matrix notation, the equations (3.2) are written as

$$S = T^T T, \tag{3.3}$$

where S is the $n \times n$ matrix whose (i, j)-element is $s(\mathbf{t}_i, \mathbf{t}_j)$ and T is the $n \times n$ matrix whose i-th column vector is $\boldsymbol{\tau}_i$. Here, we perform the eigendecomposition[14]. Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of S such that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n. \tag{3.4}$$

Let $\mathbf{a}_i$ be the eigenvector corresponding to λ_i. We define the matrix L as

$$L = \mathrm{diag}(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_n}), \tag{3.5}$$

and the matrix A as the $n \times n$ matrix whose i-th column vector is $\mathbf{a}_i$. Then, S can be decomposed as follows:

$$S = ALL^T A^T. \tag{3.6}$$

From (3.3) and (3.6), T can be obtained as

$$T = L^T A^T. \tag{3.7}$$

So, the mapped samples can be analytically computed by the eigenvalues and the eigenvectors of S.

3.3 Cone Cluster Extraction

3.3.1 Seeking Cluster Center

As the second phase of the ARC method, we describe the cone cluster extraction method. In this method, the center $\boldsymbol{\mu} \in \Re^n$ of the cone cluster is determined so that the function

$$p_c(\boldsymbol{\mu}) = \sum_{i=1}^{n} w_c(d_c(\boldsymbol{\mu}, \boldsymbol{\tau}_i)) \tag{3.8}$$

is minimized, where the cone distance is described as

$$d_c(\boldsymbol{\mu}, \boldsymbol{\tau}_i) = 1 - \frac{\boldsymbol{\mu}^T \boldsymbol{\tau}_i}{\| \boldsymbol{\mu} \| \| \boldsymbol{\tau}_i \|}, \tag{3.9}$$

and the window function $w_c(x)$ is described as

$$w_c(x) = (1 - \frac{x}{\nu})^2, \tag{3.10}$$

and ν is the parameter which determines the size of the window function.

We will show the optimal solution can be computed analytically, when the size of the window function ν is fixed to 1. First, the norm of the samples $\boldsymbol{\tau}_i$ is normalized to one,

$$\boldsymbol{\tau}_i \leftarrow \frac{\boldsymbol{\tau}_i}{\| \boldsymbol{\tau}_i \|}. \tag{3.11}$$

By this normalization, the optimal cluster center does not change, because the function $p_c(\boldsymbol{\mu})$ is not affected at all. Also, since the norm of $\boldsymbol{\mu}$ does not affect $p_c(\boldsymbol{\mu})$, we assume that $\| \boldsymbol{\mu} \| = 1$. Then, the optimization problem is rewritten as follows: Find $\boldsymbol{\mu}$ that maximize

$$\sum_{i=1}^{n} (\boldsymbol{\mu} \boldsymbol{\tau}_i)^2 \tag{3.12}$$

subject to the constraint

$$\| \boldsymbol{\mu} \|^2 = 1. \tag{3.13}$$

Let J be the $n \times n$ correlation matrix[25] of the samples:

$$J = \sum_{i=1}^{n} \boldsymbol{\tau}_i \boldsymbol{\tau}_i^T \tag{3.14}$$

Then, the optimization problem can be rewritten as: Find $\boldsymbol{\mu}$ that maximize

$$\boldsymbol{\mu}^T J \boldsymbol{\mu}, \tag{3.15}$$

subject to the constraint

$$\| \boldsymbol{\mu} \|^2 = 1. \tag{3.16}$$

The optimal solution of the problem is given by the following theorem[26].

Theorem 1 *The optimal solution of the problem of (3.15) can be obtained as*

$$\boldsymbol{\mu} = \mathbf{z}_1, \tag{3.17}$$

where $\mathbf{z}_1$ is the first eigenvector (i.e. the eigenvector corresponding to the largest eigenvector) of J.

(proof)

When the constraint is taken into account by the use of Lagrange multiplier γ, (3.15) is rewritten as

$$Q(\boldsymbol{\mu}) = \boldsymbol{\mu}^T J \boldsymbol{\mu} - \gamma(\boldsymbol{\mu}^T \boldsymbol{\mu} - 1). \tag{3.18}$$

The gradient vector of $Q(\boldsymbol{\mu})$ is denoted as

$$\nabla Q(\boldsymbol{\mu}) = 2J\boldsymbol{\mu} - 2\gamma\boldsymbol{\mu}. \tag{3.19}$$

The stationary points of Q are obtained as the solution of $\nabla Q(\boldsymbol{\mu}) = 0$. So, the stationary points of Q satisfy the following equation:

$$J\mathbf{m} = \gamma\mathbf{m}. \tag{3.20}$$

Thus, the stationary points of Q are given by the eigenvectors of J. Let $\nu_1, \cdots, \nu_n$ be the eigenvalues of J where

$$\nu_1 \geq \nu_2 \geq \cdots \geq \nu_n \tag{3.21}$$

and let $\mathbf{z}_i, \cdots, \mathbf{z}_n$ be the eigenvectors of J. The maximum point is given by the stationary point with the largest Q. Since $Q(\mathbf{z}_i) = \nu_i$, $Q(\boldsymbol{\mu})$ is maximized when $\boldsymbol{\mu} = \mathbf{z}_1$. □

3.3.2 Sequential Extraction of Cone Clusters

In the cone cluster extraction method, the clusters are extracted one by one. First of all, the first cluster center $\boldsymbol{\mu}_1$ is determined. Then, the samples close to the center are extracted as a first cluster $\mathcal{C}_1$. The samples extracted as the cluster are removed from the sample set, and the same procedure is repeated again to extract the second cluster $\mathcal{C}_2$. The iteration continues until the sample set becomes empty or the predetermined number of clusters are extracted.

We will explain how to choose the samples in the cluster $\mathcal{C}_k$. First, we define the membership value of the sample $\boldsymbol{\tau}_i$ to $\mathcal{C}_k$ as follows:

$$u_{ki} = w_c(d_c(\boldsymbol{\mu}, \boldsymbol{\tau}_i)). \tag{3.22}$$

Basically, the samples whose membership values are above the threshold are chosen as the cluster. However, since the size of a cluster is various, it is not appropriate to set a common threshold for every cluster. So, the samples of the cluster $\mathcal{C}_k$ are chosen by the following algorithm. Initially, $\mathcal{C}_k$ is a null set. The samples are added to $\mathcal{C}_k$ one by one in descending order of u_{ki}. The adding process continues until

$$\sum_{\mathbf{t}_i \in C_k} u_{ki} > \eta \sum_{i=1}^{n} u_{ki}, \tag{3.23}$$

where η is the parameter which determines the cluster boundary.

3.4 Experiments on Robustness against Noise

In this section, the ARC method is applied to the artificial data which contain a number of noises. The robustness against noise of the ARC method is compared with those of C-Means and Noise-Resistent C-Means(NR C-Means)[11].

We produced 500 clustered samples and 300 noise samples on the two dimensional region $\{(x, y)|0 \leq x \leq 1, 0 \leq y \leq 1\}$. The clustered samples are placed at 7 positions using Neyman-Scott Process[1]. The procedure of Neyman-Scott Process is as follows:

1. Determine the number of samples contained in each cluster.
2. Set the central position of each cluster so that the distance between every two centers is more than 0.2.
3. Produce samples of each cluster so that the distances from the samples to the center have the normal distribution with expectation 0 and variance α^2.

We prepared five kinds of test data with $\alpha = 0.02, 0.04, 0.06, 0.08, 0.1$. The number of samples of each cluster is determined randomly so that each cluster contains more than 5% of all samples. The noise samples are scattered over the region according

to the uniform distribution. This kind of test data is frequently used for evaluating the robustness of the clustering algorithms against noise[27, 11].

In all methods, the number of clusters is set to the true number of clusters in advance. In C-Means and NR C-Means, the initial cluster centers are determined as follows:

1. The first center is chosen as the nearest sample to the arithmetic mean of all samples.
2. Other centers are randomly chosen from the samples so that the distance between every two centers is more than 0.2.

Ten trials are made and the most successful one which achieved the minimum value of the error function is selected.

In the evaluation of results, the obtained cluster $\mathcal{C}_k(k = 1, \cdots, c)$ are matched against the prepared cluster $\mathcal{B}_j(j = 1, \cdots, c)$. The cluster $\mathcal{C}_k$ is judged as correctly extracted, if

- There is a prepared cluster $\mathcal{B}_j$ in which the 90% of the non-noise samples of $\mathcal{C}_k$ are contained.
- No other cluster $\mathcal{C}_i(i \neq k)$ satisfies the first condition.

In C-Means and NR C-Means, each sample is assigned to the cluster which has the largest membership value. We call the ratio of the number of the correctly extracted clusters to that of prepared ones "the extraction rate".

In the ARC method, the parameters are set as follows: $\eta = 0.9, \sigma = 0.06$. In NR C-Means, the value of δ is set empirically at 0.12 as follows: the clustering is performed at $\delta = 0.1, 0.11, ..., 0.2$, and we chose the value that achieved the largest sum of the extraction rates at $\alpha = 0.02, 0.04, \cdots, 0.1$.

The average extraction rate over 15 trials is shown in Fig.3.4. As α increases, the duplication between clusters increases. So, dividing into clusters becomes hard and the extraction rate decreases. The performance of C-Means is much worse than

the other two, because C-Means does not assume the existence of noise. When α is large, the ARC method performs better than NR C-Means significantly. The result shows thet the ARC method has higher robustness than NR C-Means.

The example of clusters produced by the ARC method is shown in Fig. 3.2 ($\alpha = 0.06$). The mark $\square$ shows the cluster obtained at $\eta = 0.99$. You can see that the noises are excluded and only high density areas are extracted. It must be noted that the clusters extracted by our method are spherical. When a cluster is not spherical, it is extracted as a sum of several clusters (Fig. 3.3).

3.5 Summary

In this chapter, we described the ARC method which is implemented by the combination of the inner product scaling and the cone cluster extraction. We compared the noise robustness of the ARC method with that of NR C-Means method in the experiment with artifitially generated samples, and found that the ARC method is more robust to noises than NR C-Means. The suitable applications of the ARC methods are considered to be the clustering where a lot of noises are included. The following three chapters present the three applications of the ARC method.

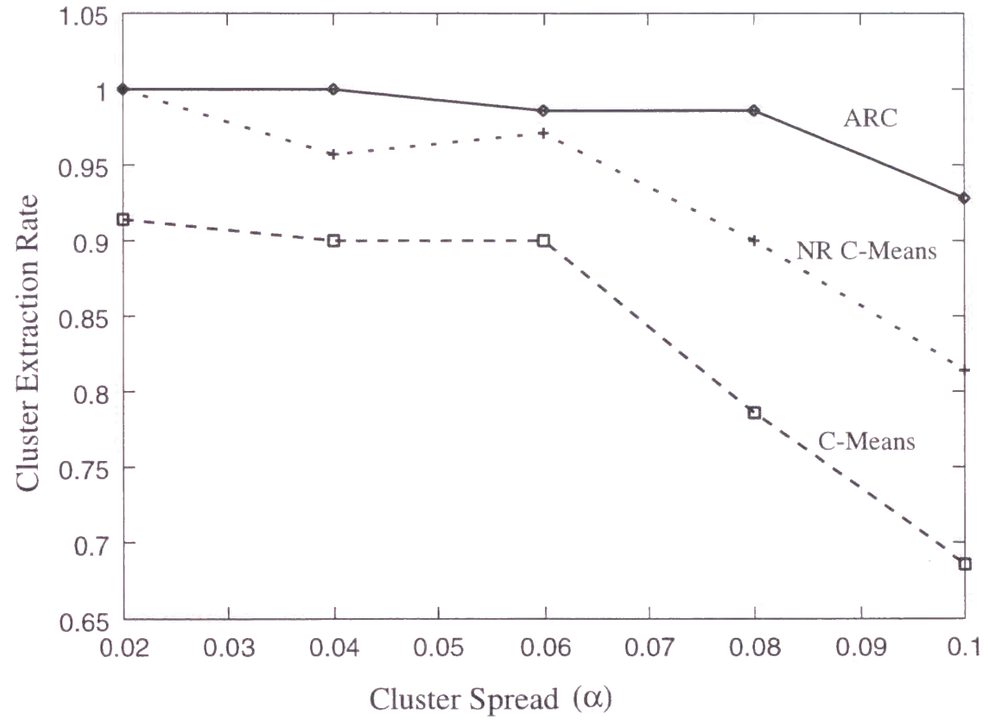


Figure 3.1: Cluster extraction rates of the ARC method, Noise-Resistent C-Means and C-Means on noisy data

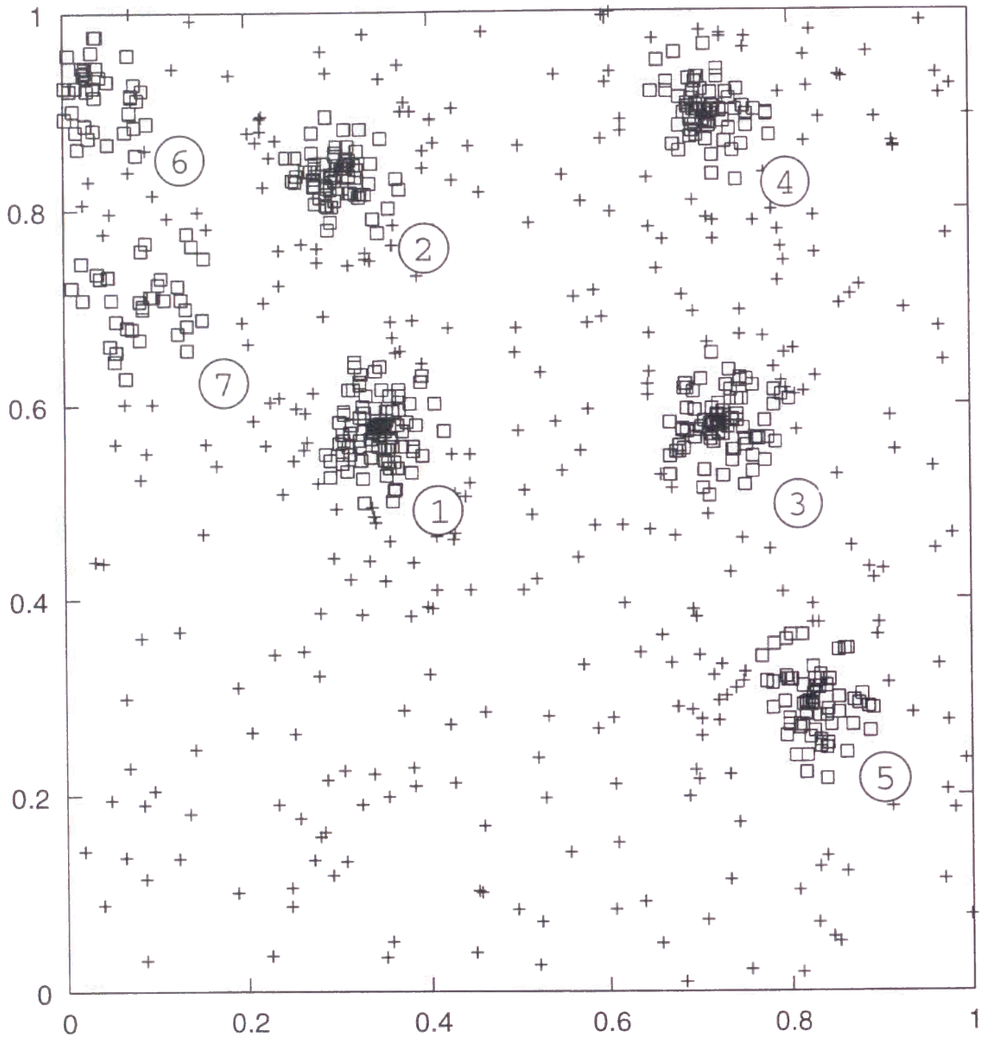


Figure 3.2: An example of cluster extraction by the ARC method (1)

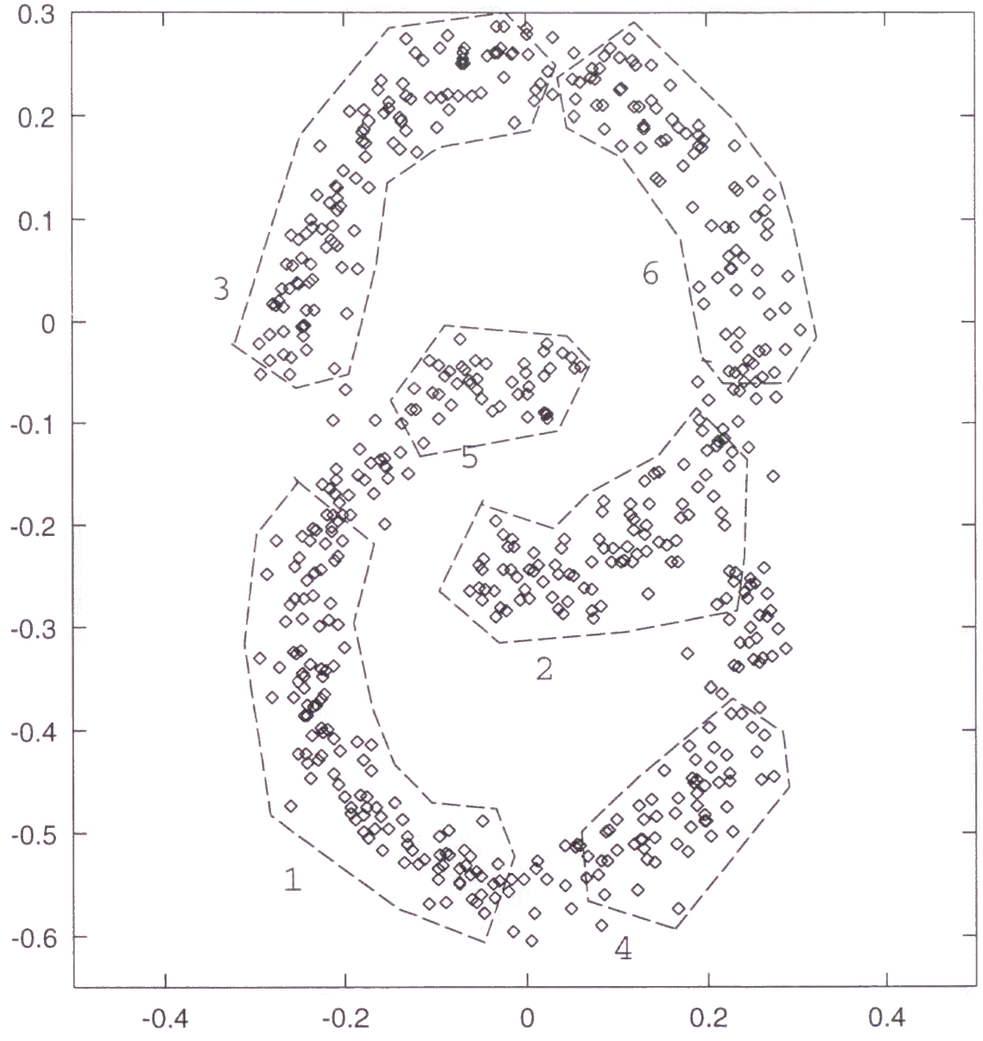


Figure 3.3: An example of cluster extraction by the ARC method (2)

Chapter 4

Application: Extracting Lines from an Image

4.1 Introduction

Extraction of lines from an image is an essential task in computer vision. Many authors have reported various algorithms for line extraction, such as Hough Transformation and gradient-based methods[28]. Among them, clustering line segments[29, 30, 31, 32] is an effective approach for line extraction. A line segment is a small geometric structure defined by two end points. Line segments are obtained by applying the line fitting process to the output of the edge detection process. In many cases, line segments lie along long straight lines. By clustering these segments, straight lines in the image are extracted and the large linear structure can be found.

The line extraction process usually consists of two independent phases. First, the "collinearity" (or similarity) of two line segments is measured based on their directions and on the distance between their end points. Second, the clustering of line segments is carried out. In our research, we will use the ordinal measure of collinearity and focus our attention on the second part - the clustering algorithm.

As a clustering problem, the line extraction has the following three characteristics:

- The similarity between the samples is defined instead of the distance.
- Many noises are contained in the sample set.
- The number of samples and the number of clusters are very large.

The ARC method is considered to be appropriate for the line extraction, since the characteristics are desirable for the method.

For the first characteristic, the ARC method is straightforwardly adaptable to the similarity data. In the ARC method, the Gaussian similarity is used to perform the inner product scaling as in (3.1). When you would like to perform clustering using another similarity, the Gaussian similarity should be simply replaced by the similarity. For the second characteristic, it has high robustness against noise. For the third characteristic, the property that clusters can be obtained by analytical computations is suitable for large scale problems. The partitional methods such as NR C-Means might not be appropriate because the risk of stucking at local minima is very high when the number of samples and clusters are very large. In this chapter, we describe the application of the ARC method to line extraction, and its advantages over the previous methods are discussed.

The rest of this chapter is organized as follows: In Sec. 4.2, previous methods are reviewed and their problems are pointed out. In Sec. 4.3, the similarity measure between the line segments is described. In Sec. 4.4, it is described how a cluster of line segments is replaced by a longer line. In Sec. 4.5, we describe the experiments where the ARC method is applied to a synthetic image, a natural image, and a rough sketch. In Sec. 4.6, the line extraction process is extended to cope with arbitrary 2D shapes. We will also show several preliminary experiments on 2D shapes. Finally, we conclude our discussion in Sec. 4.7.

4.2 Previous Clustering Methods for Line Extraction

Let us review some of the previous clustering methods for the line extraction task. The two well-known methods are the thresholding method[29] and the hierarchical line clustering method[30].

In the thresholding method, two line segments whose collinearity is above a certain threshold are considered to constitute the same line. A cluster is formed by traversing the segments whose collinearities are above the threshold. The advantage of the method is its simplicity and high speed. The time complexity of this algorithm is $O(n^2)$ and the storage requirement is $O(n)$, where n is the number of line segments.

In the hierarchical line clustering method, it is repeated that the most collinear pair of lines are merged into a longer line by the procedure described in Sec. 4.4. The output of the algorithm is a binary tree whose terminal nodes are the original line segments. Each non-terminal node is a line segment produced by merging of its two children. Lines can be obtained by cutting this tree at a level where the collinearity between lines are less than a threshold. This is a very slow method; the time complexity of this algorithm is $O(n^3)$ and the storage requirement is $O(n)$. Its advantage over thresholding method is that the threshold can be determined after grouping.

These two methods share one characteristic in common: they both use a threshold to obtain clusters. The main difference between them is when thresholding is conducted: before or after grouping. In these methods, the proper setting of the threshold is very important to obtain desired results. If the threshold is too high, the line segments will be divided into many clusters so that the extraction is almost meaningless. On the other hand, if the threshold is too low, two kinds of problems will occur: *overclustering* and *noise inclusion*. The overclustering is the phenomenon that two obviously distinct lines are merged into one. On the other hand, the noise inclusion is the one that the "noise segments" are included in the extracted lines.

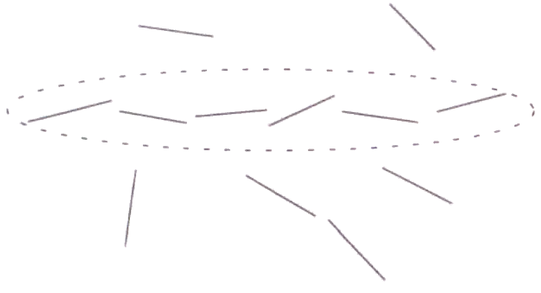


Figure 4.1: An example of weakly-collinear line segments with noise segments

The noise segments or "noises" are the randomly-oriented short segments produced by the line segment fitting process. For example, if the image contains textured areas (e.g. trees in scene images), these areas will be filled with noises. If noise segments are by chance collinear and the collinearity between them is above the threshold, they are clustered as a straight line. Noise inclusion emphasizes trivial lines that do not belong to the structure of the contents in the image.

Consider the situation shown in Fig. 4.1. The line segments comprising a line are weakly collinear, because the directions of line segments deviate from the true direction of the original line during the edge extraction, thinning and line fitting. There are several noise segments and some of them are collinear by chance.

Since the collinearity of the weakly-collinear segments is small due to the difference of directions, the threshold must be low to extract these segments as a line. However, if the similarity between the noise segments are above the threshold, the noise inclusion occurs. In such cases, it is impossible to cluster weakly-collinear line segments as a one line by a threshold.

4.3 Similarity between Line Segments

Since the line segment is characterized by the two end points in the two dimensional space, the line segment is described as the four dimensional vector $\mathbf{l}_i$. The similarity

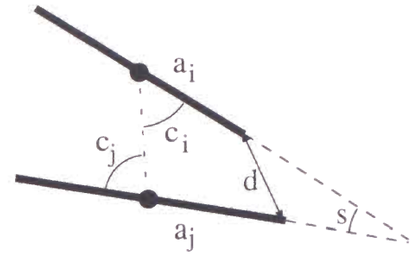


Figure 4.2: Parameters of the Measure of Collinearity

between two line segments is determined by the parameters shown in Fig.4.2[33]. The similarity value between the two segments $\mathbf{l}_i$ and $\mathbf{l}_j$ is

$$s(\mathbf{l}_i, \mathbf{l}_j) = \frac{a(\zeta_d - d/a)(\zeta_s - s)(\zeta_c - c)}{\zeta_d \zeta_s \zeta_c}, \tag{4.1}$$

where a is the average of a_i and a_j, the lengths of two segments, d is the distance between the nearest end-points, s is the difference of angles between two segments, c is the average of c_i and c_j, which are the angles between the line connecting the centers, ζ_d, ζ_s and ζ_c are the constants that determine the effective ranges of these variables.

4.4 Replacing Line Segments by a Single Line

When the clusters of line segments are obtained, each cluster has to be replaced by a single straight line. The straight line replacing them is obtained as follows; First, the straight line that minimizes the mean squared error of all the end points to the line is constructed. Then, all the segments are projected to the line. The end points of the straight line are determined as the two most distant points among the end points of the projected segments. In the hierarchical line clustering, this replacement procedure is used to merge two segments in the course of growing clusters. In the ARC method, this replacement procedure is used after all clusters

are found. However, you may replace each cluster one by one right after it is found, since the clustering procedure and the replacement procedure are independent.

4.5 Line Extraction Experiments

4.5.1 Synthetic Image

With respect to the ability to extract weakly-collinear segments, the ARC method and the hierarchical line clustering were compared using the synthetic image shown in Fig. 4.3. There is a long line divided into 13 weakly-collinear segments in the center of the image, and 150 widely scattered noise segments. The lengths of the line segments were set to the same value. The direction of the segments belonging to the line differs slightly from its true direction. The difference of the direction of each segment was set randomly from $-7°$ to $7°$. The parameters were set as follows: $\zeta_d = 5, \zeta_s = 20°, \zeta_c = 20°$. The task is to extract this long line without extracting noise segments.

When the hierarchical line clustering was used with the threshold of 10 (Fig. 4.4(a)), the long line was broken into three pieces and four noise inclusions occured. If you make the threshold higher, the noise inclusions can be resolved but the broken lines will never be extracted as the line. For example, when the threshold was 14 (Fig. 4.4(b)), the noise inclusions decreased but the long line was broken into more pieces. Whereas, if you make the threshold lower, the long line will be extracted as the line, but the noise inclusions will never be reduced. For example, when the threshold was 8 (Fig. 4.4(c)), the long line was extracted as the line, but the noise inclusions increased. In this synthetic image, the noise inclusion could not be avoided by adjusting the threshold without breaking the long line into pieces.

In contrast, in the ARC method, the noise inclusion could be avoided (Fig. 4.4(d)). The long line was extracted as the first cluster, and so no other lines were extracted from noise segments. This result suggests that the ARC method is superior to the hierarchical line clustering in extracting weakly-collinear segments.

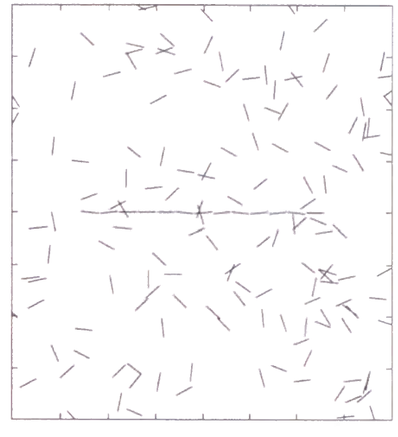


Figure 4.3: Synthetic Image with Noise

4.5.2 Textured Image

The line extraction was applied to a natural scene. The image used for the experiment is the snapshot of a square paper placed on small pebbles(Fig.4.5(a)). The purpose of the experiment is to extract the four lines that correspond to the boundary of the paper. In the line extraction tasks from natural scenes, such textured images are considered to occur frequently.

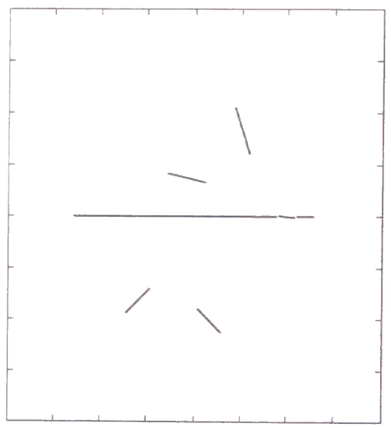
Edge detection, binarization, thinning and line segment fitting are performed on the image. The obtained line segments are shown in Fig. 4.5(b). The small line segments whose length is less than 3 pixels are omitted. The texture of pebbles produces the noisy line segments.

The similarity between the line segments $\mathbf{l}_i$ and $\mathbf{l}_j$ is determined as follows:

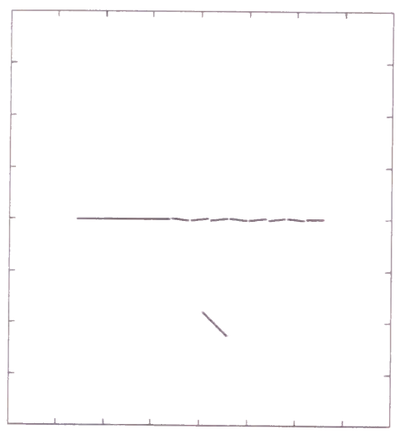
$$s(\mathbf{l}_i, \mathbf{l}_j) = \exp(-\frac{d}{\xi_d} - \frac{s}{\xi_s} - \frac{c}{\xi_c}), \tag{4.2}$$

where ξ_d, ξ_s, ξ_c are the weights. Here, the weights are set as follows: $\xi_d = 80$(pixels), $\xi_s = \xi_c = 10°$.

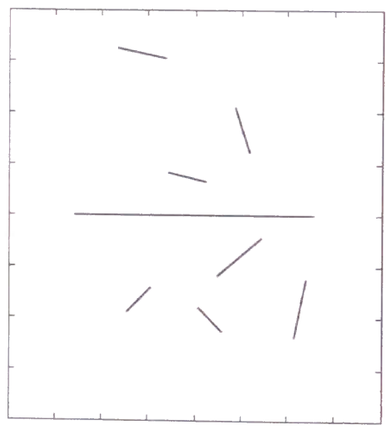
Fig. 4.5(c) shows the extracted lines by the ARC method. You can see the four lines of boundary are almost completely extracted. On the other hand, Fig.



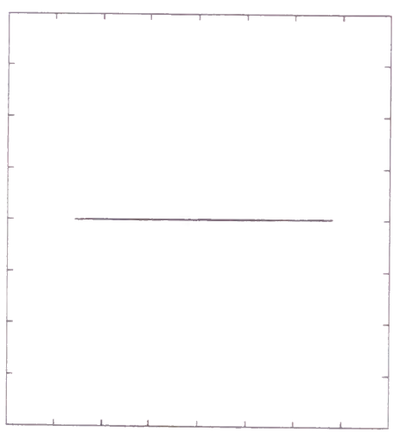
(a) Extracted lines by the hierarchical clustering (threshold = 10)



(b) Extracted lines by the hierarchical clustering (threshold = 14)



(c) Extracted lines by the hierarchical clustering (threshold = 8)



(d) Extracted lines by the ARC method

Figure 4.4: Results for a synthetic image with noise

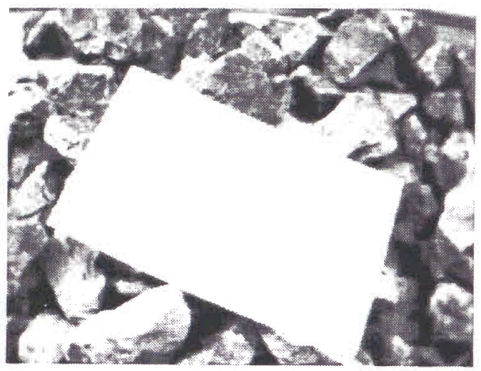
4.6(a)(b)(c) show the extracted lines by the hierarchial clustering with the threshold of 0.2, 0.4, 0.6, respectively. In the case of low threshold, the four lines are extracted, but many noise lines are also extracted. On the other hand, in the case of high threshold, the four lines are not extracted at all. You can see that the four lines cannot be extracted without extracting noise lines simply by the adjustment of threshold. This result suggests the effectiveness of the the ARC method in image processing tasks.

4.5.3 Rough Sketch

The application of the line extraction is not limited to computer vision. The line extraction is applicable to reforming a rough sketch[33]. A rough sketch is a kind of record of the design process and so several ideas are presented at the same time in a sketch. A rough sketch has a characteristic that several lines are drawn to represent a line. Our primary objective is to cluster such lines into the line which the designer wanted to draw. This clustering can help the designer summarize his ideas. We call this task the *rough sketch reforming*.

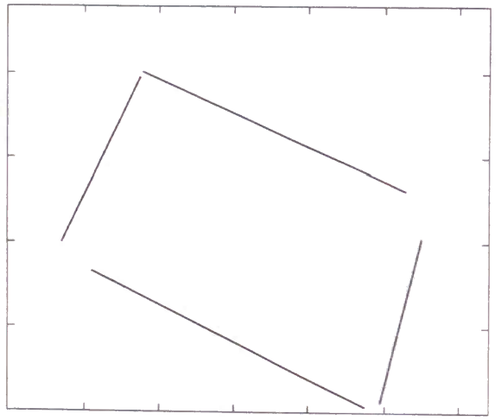
In this experiment, both the ARC method and the hierarchical line clustering were applied to an on-line rough sketch. Since this sketch of a car is drawn on an on-line environment, tracks of the pen can be obtained. Line fitting process is applied to these tracks instead of the pixels in the image. In the on-line environment, no noise segments are caused by edge extraction process and thinning process[33].

The original image contains 2315 line segments (Fig.4.7). 100 straight lines were extracted by the ARC method (Fig.4.8(a)) and there were 1149 line segments that remained unclustered. The reformed result was made by joining the extracted straight lines and the unclustered line segments (Fig. 4.8(b)). Therefore, the number of the lines in the reformed result was 1249. The parameters were set as follows; $\zeta_d = 1, \zeta_s = 12°, \zeta_c = 8°$. In the reformed result of the ARC method, overlapped strokes were clustered into a single line and the sketch was simplified as a whole. However, several overclustered segments were found (e.g. in the wheels of the car).



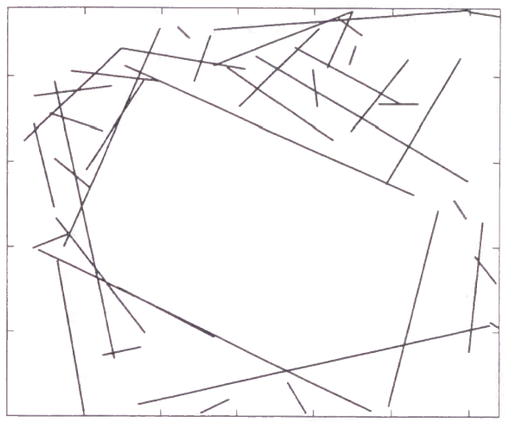
(a) Original image

(b) Line segments fitted to the edge

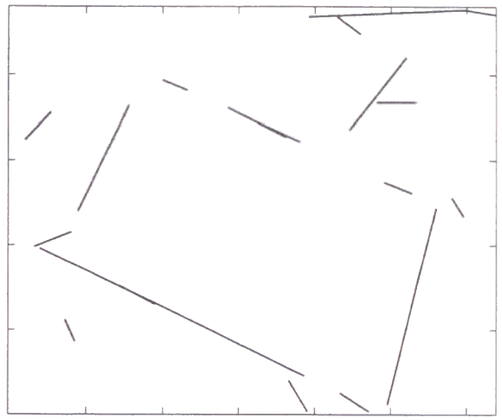


(c) Lines extracted by the ARC method

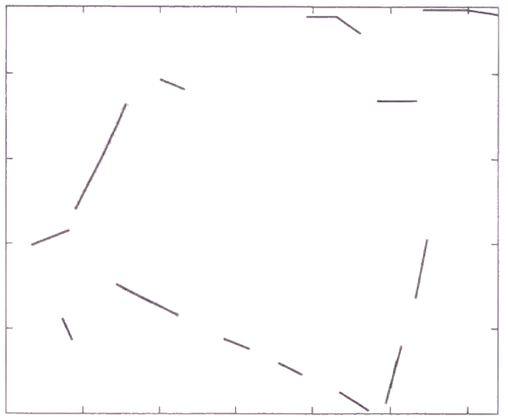
Figure 4.5: Extracted straight lines from a textured image by the ARC method



(a) Lines extracted by the hierarchical clustering (threshold=0.2)



(b) Lines extracted by the hierarchical clustering (threshold=0.4)



(c) Lines extracted by the hierarchical clustering (threshold=0.6)

Figure 4.6: Extracted straight lines from a textured image by hierarchial clustering

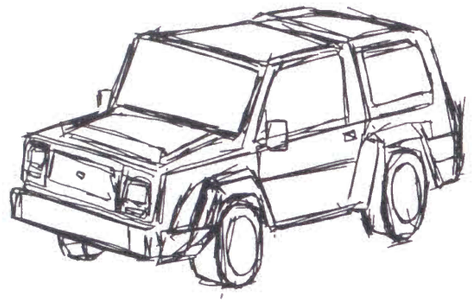


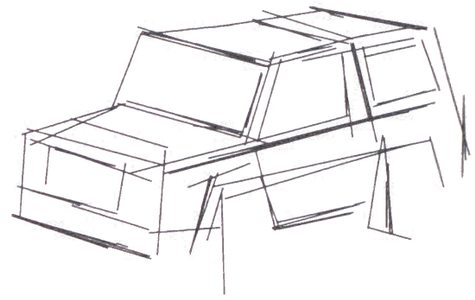
Figure 4.7: Original Sketch (2315 lines)

The reformed result by the hierarchical line clustering is shown in Fig.4.8(c). The number of lines in the reformed result of the hierarchical clustering is equal to that of the ARC method. We can visually compare the two methods on the occurence frequency of overclustering, since the difference is obvious: the number of overclustering caused by the ARC method is less than that of the hierarchical clustering. To reduce the overclustering in the hierarchical line clustering method, the threshold must be set lower, which will then sacrifice the reforming effect. This result shows that the ARC method outperforms the hierarchical line clustering on reforming the rough sketch.

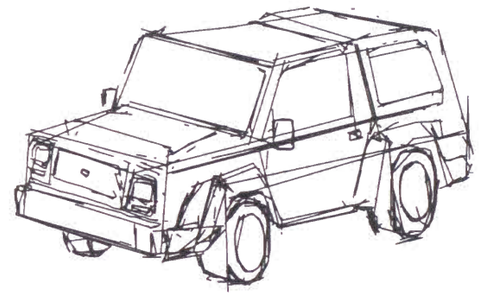
4.6 Rotation Invariant 2D Figure Extraction

By several modifications on the similarity computations, arbitrary 2D figures can be extracted by the ARC method. In this section, we will show a simple example of extracting 2D figures.

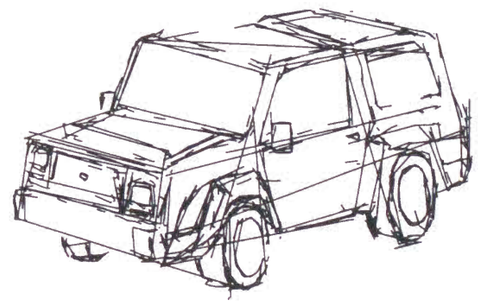
We assume that the image is represented by a set of line segments like the cases of line extraction. The template of the figure is also represented as a set of line segments. The similarity between the two line segments in the image is determined according to the probability that the two segments simultanously belong to the



(a) Extracted lines of the ARC method



(b) Reformed result of the ARC method (1249 lines)



(c) Reformed result of the hierarchical clustering (1249 lines)

Figure 4.8: Line extraction from a rough sketch

figure of the template. Then, the ARC method is performed, and the figures are extracted as clusters of line segments.

This method has the drawback that the number of line segments in the template must be equal to that of the figure in the image. Therefore, this method is not scale-invariant. But, this method is rotation-invariant, since the relational position of line segments is used for matching.

4.6.1 Similarity based on Template

In 2D figure extraction, the similarity should be determined based on the probability that the two segments simultanously belongs to the figure. First, five basic features are defined on the two line segments (Fig. 4.9).

θ : The angle between two line segments.

$a1$: The distance between the crossing point K and $A1$ (i.e. the nearer end point of line segment A).

$a2$: The distance between the crossing point K and $A2$ (i.e. the farther end point of line segment A).

$b1$: The distance between the crossing point K and $B1$ (i.e. the nearer end point of line segment B).

$b2$: The distance between the crossing point K and $B2$ (i.e. the farther end point of line segment B).

Let the line segments of the template be $\mathbf{L}_p(p = 1, \cdots, m)$, and the feature values of $\mathbf{L}_p$ and $\mathbf{L}_q$ be Θ_{pq}, $A1_{pq}$ and so on. Let the line segments of the image be $\mathbf{l}_i(i = 1, \cdots, n)$, and the feature values of $\mathbf{l}_i$ and $\mathbf{l}_j$ be θ_{ij}, $a1_{ij}$ and so on. The similarity between $\mathbf{l}_i$ and $\mathbf{l}_j$ is obtained as follows:

$$s(\mathbf{l}_i, \mathbf{l}_j) = \max_{p \neq q, p=1\cdots m, q=1\cdots m} sim(\mathbf{l}_i, \mathbf{l}_j, \mathbf{L}_p, \mathbf{L}_q), \quad (i, j = 1, \cdots, n), \tag{4.3}$$

where

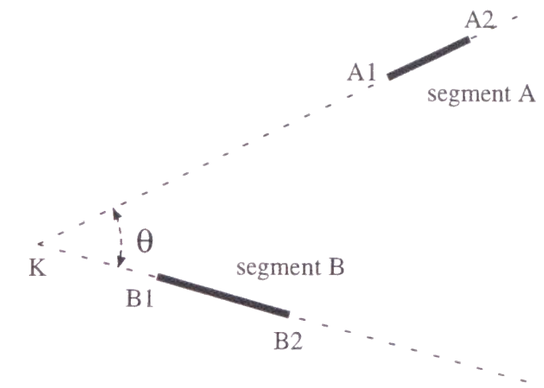


Figure 4.9: Similarity for 2D Figure Extraction

$$sim(\mathbf{l}_i, \mathbf{l}_j, \mathbf{L}_p, \mathbf{L}_q) = v(a1_{ij}, a2_{ij}, A1_{pq}, A2_{pq}) v(b1_{ij}, b2_{ij}, B1_{pq}, B2_{pq})(\theta_t - |\theta_{ij} - \Theta_{pq}|)$$

and

$$v(a1, a2, A1, A2) = \begin{cases} 0 & (a1 > A2) \text{ or } (a2 < A1) \\ 1 & (a1 \geq A1, a2 \leq A2) \\ (A2 - a1)/(A2 - A1) & (A1 \leq a1 \leq A2, a2 > A2) \\ (a2 - A1)/(A2 - A1) & (A1 \leq a2 \leq A2, a1 < A1) \\ (A2 - A1)/(a2 - a1) & (a1 \leq A1, a2 \geq A2) \end{cases}.$$

The similarity $s(\mathbf{l}_i, \mathbf{l}_j)$ is determined as the maximum value of $sim(\mathbf{l}_i, \mathbf{l}_j, \mathbf{L}_p, \mathbf{L}_q)$, which is defined as the difference of the angle and the degree of overlap v of the line segments. The variable $sim(\mathbf{l}_i, \mathbf{l}_j, \mathbf{L}_p, \mathbf{L}_q)$ has a large value when the angle difference is small and the overlap is large. The parameter θ_t is a threshold of the angle difference, which was set to 3^o in this experiment.

4.6.2 Experimental Result

We extracted the template figure shown in Fig. 4.11(b) from the image in Fig. 4.11(a). The image and the template were respectively divided into line segments such that the length of every line segment is 10 dots (Fig. 4.11(c)). Then, the

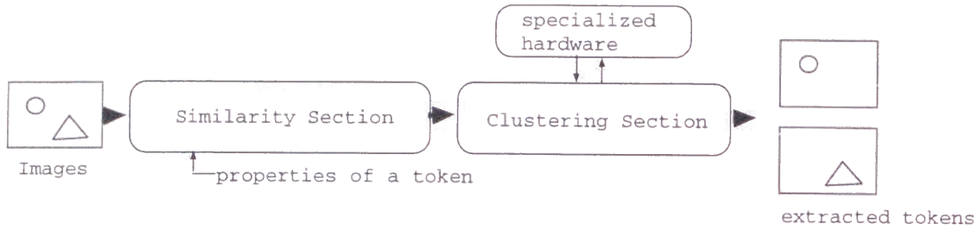
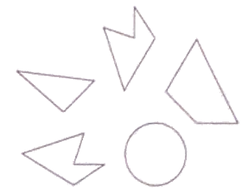


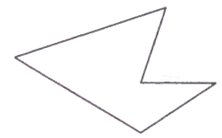
Figure 4.10: Clustering-based Token Extraction System

number of line segments in the image was 160 and that of the template was 32. The similarities were set between every two line segments in the image. They were normalized so that the maximum value becomes 1. Lines in Fig. 4.11(d) indicate the similarity values between the line segments are more than 0.2. You can see that there are high similarities between the line segments that consists the figures which match the template. We extracted two clusters using the ARC method and the results are shown in Fig. 4.11(e),(f). You can see that the correct figures are extracted regardless of the rotation.

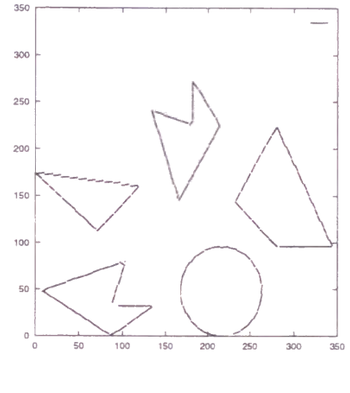
The basic element of images are called "tokens". Tokens include regions, edge, lines and so on. So far, many specialized methods are proposed for each kind of the tokens, but there is no method that can deal with all tokens. Since all tokens can be considered as clusters of pixels, the ARC clustering method has a possibility to provide a unified framework for extracting various tokens. The overview of the clustering-based token extraction system is shown in Fig. 4.10. This system consists of two sections. The similarity section calculates the similarities between pixels according to the properties of a particular token. Then, the clustering section produces clusters that correspond to the tokens. Since the clustering section is independent from the similarity section, this part can be implemented by hardware.



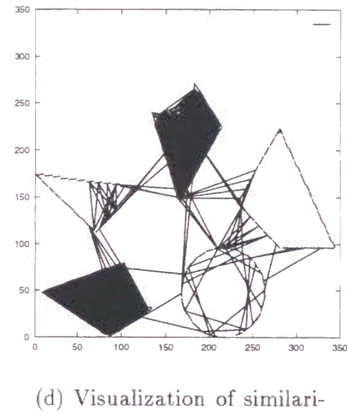
(a) Image



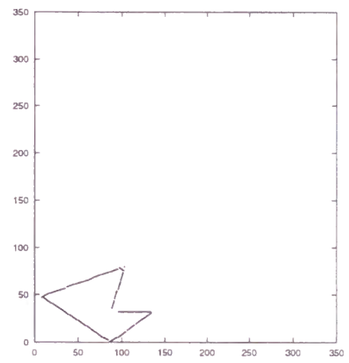
(b) Template



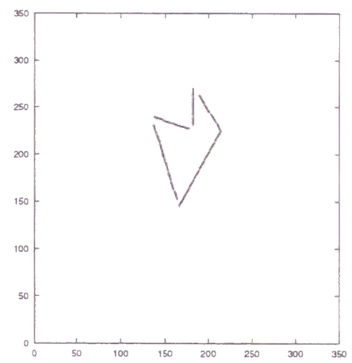
(c) Fitted line segments



(d) Visualization of similarities



(e) The figure extracted as the first cluster



(f) The figure extracted as the second cluster

Figure 4.11: Extraction of 2D figures by the ARC method

4.7 Summary

In this chapter, we applied the ARC method to the line extraction process. As a result, weakly-collinear line segments are successfully clustered as a line, which was very difficult for the conventional threshold-based methods. We have applied the method to rough sketch reforming and confirmed that it outperforms the hierarchical line clustering method. The ARC method performed well also in the line extraction from a natural textured image. The ARC method could be applied to other extraction tasks from an image even when many noises are included.

Chapter 5

Application: Clustering-based Browsing of Document Database

5.1 Introduction

In general, there are two ways to access a document database; searching and browsing. In searching, a user makes a query with keywords and the retrieval system returns the documents related to the query. In browsing, the user examines the whole database and looks for the documents that match his interests. If the user has a definite idea of what he needs, choosing keywords is an easy job. If not, the user needs to browse over the database first to make his needs more clear. Browsing function is indispensable for those who have vague needs.

In the vector space representation[5], a document is represented by a vector of the term occurence frequency. Let the whole document database has q unique terms, then terms can be indexed from 1 to q. Since there are so many terms as listed in the dictionary, q is usually very large ($10^3 \sim 10^4$). A document is represented by a q-dimensional feature vector, where the i-th element denotes the number of occurence of term i in the document (Fig. 5.1). In document clustering, the normalized cosine[5] is used for measuring the similarity between the documents.

So far, document clustering is mainly used to increase the efficiency of keyword

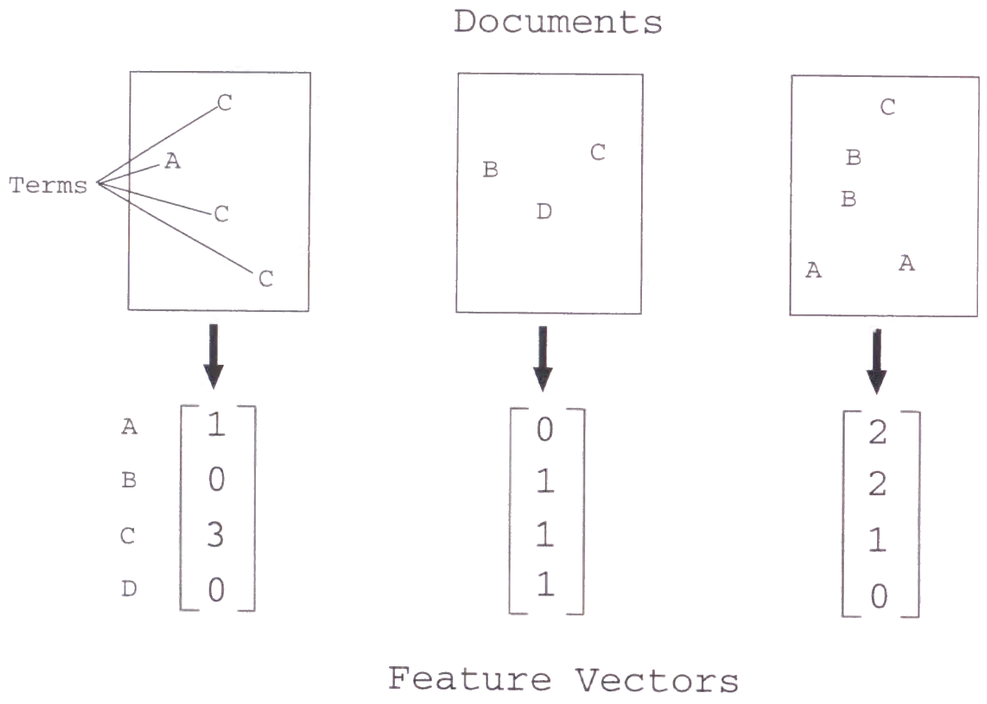


Figure 5.1: Vector space representation of documents

search[34, 24]. The purpose of the keyword search is to find the documents that contain user-specified keywords. To make the keyword search efficient, the number of examined documents should be reduced. First, the documents are divided into clusters and the vectors of the documents of each cluster are summed into one vector. If user-specified keywords are not indicated in the summed vector of a cluster, you need not to refer the documents in the cluster, which reduces the number of examined documents effectively.

Document clustering is recently used as a powerful tool for browsing [15]. An outline of cluster-based browsing system is shown in Fig. 5.2. The documents are indexed by the term occurrence frequency and pairwise similarities are set according to the number of common terms. Similar documents are clustered and a *representative document* of each cluster is shown to the user. The user can get the whole view of the database without examining documents one by one.

There are two requirements for a document clustering method. First, the method must make "tight" clusters, in which documents are connected with high similarities. In order to infer the contents of one cluster from a representative document, the similarities between the representative and each of the other documents must be high enough. Second, the method must be fast with regard to the processing time. When the browsing system is used combined with the retrieval system, the user repeats browsing and retrieval alternately. Clustering should be done in a time tolerable for user interaction.

Most of existing document clustering methods are agglomerative methods. They repeat finding the most similar pair of clusters and merging them to make a larger cluster. Among the agglomerative methods, Complete Link method is recommended by the tightness of the clusters[24]. But its time complexity is $O(n^3)$, where n is the number of objects. In contrast, the time complexity of Single Link is $O(n^2)$, but known to make meaningless clusters[24]. None of these methods could satisfy both requirements.

In this chapter, we apply the ARC method to the document clustering in order

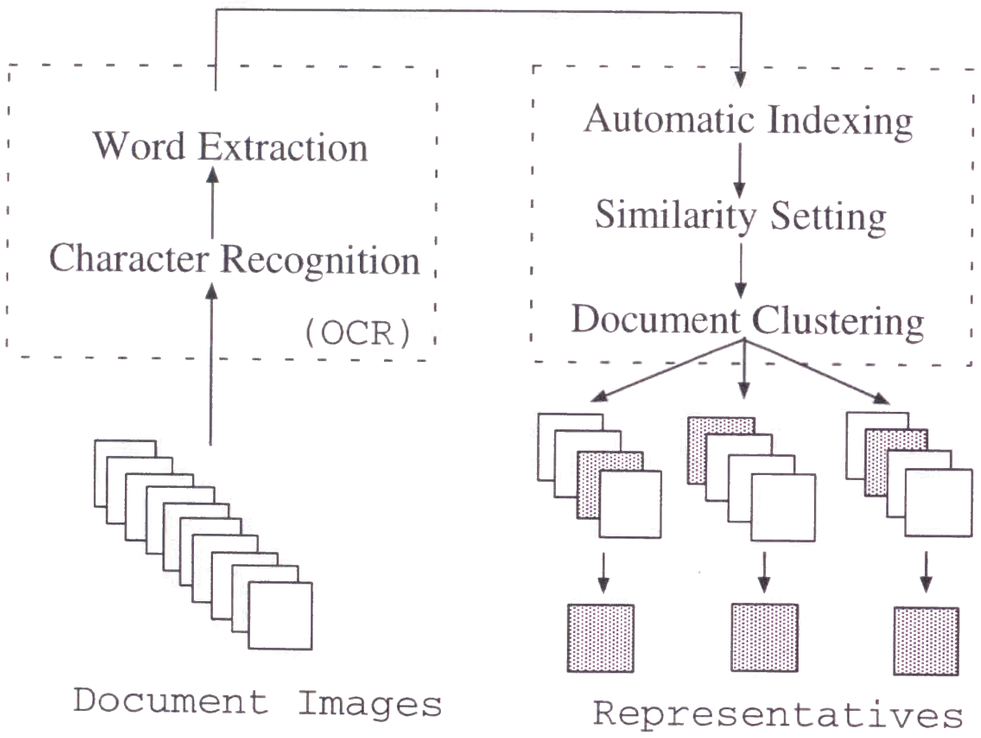


Figure 5.2: Document Image Database Browser

to construct a browsing system. Then, we will show that the ARC method can satisfy these two requirements. We examined the tightness of clusters produced by the ARC method and found our method can produce as tight clusters as Complete Link. And we compared the actual computational time of the ARC method with that of Complete Link using a workstation.

In many cases, the input to an information retrieval(IR) system is the output of an OCR system[35]. Though such OCR-ed documents contain certain amount of recognition faults, existing IR systems pay no attention to this problem. We evaluated our document clustering method with faulty documents and measured the deterioration of the tightness of clusters. As a result, when the recognition rate was more than 84.4%, our method could produce tight clusters usable for browsing. Thus, our method can be used as a tool of browsing the faulty documents.

Clustering terms also plays a great role in information retrieval systems. Since the representation of terms is very similar to that of documents, the clustering methods applicable for document clustering is also applicable for the term clustering. A term is represented by a vector whose i-th element is the number of the term contained in the i-th document. The similarity between two terms is measured by the cosine measure[5].

The term clustering is useful for reducing the number of dimensionality of the document vector. By replacing the terms by the term clusters, the dimensionality of the document vector reduces to the number of clusters. In order not to lose the retrieval accuracy, it is important that a cluster contains the terms which have mutually related meanings. When irrelevant terms are contained in a cluster, the query will retrieve many irrelevant documents, which is very annoying for users.

Since we only deal with the occurence frequencies, the semantic processing to exclude irrelevant terms is difficult. But, there is a class of irrelevant terms which can be found from the occurence frequencies. Such terms are called "trivial terms". The trivial terms are the ones that appear in documents regardless of the contents, such as "the", "degree", "important" and so on. The occurence frequency of a

trivial term is considered to be almost constant for every document. As a result, the similarities between a trivial term and other non-trivial terms tend to have a constant small value. So, the trivial terms can be considered as "noises" in a framework of clustering. Therefore, the robustness against noise of the ARC method can contribute to exclude the trivial terms from the clusters. We will evaluate the ARC method from a viewpoint of the robustness against the trivial terms.

In Sec. 5.2, the similarity measure between documents is discussed. In Sec. 5.3, it is explained how to determine the representative document in a cluster. In Sec. 5.4, the representative documents of the articles of IEEE Trans. PAMI are extracted using the ARC method. In Sec. 5.5, the performance of the ARC method is evaluated in comparison with agglomerative methods. In Sec. 5.6, the ARC method is applied to term clustering. Sec. 5.7 is the concluding remarks.

5.2 Similarity between Documents

In this section, we define the similarity between two documents. Here, we assume that the documents in the document database are indexed from 1 to n, where n is the number of documents. Also, we assume that the terms are indexed from 1 to q, where q is the number of unique terms in the document database. Let w_{ij} be the number of the term j contained in the document i.

In vector space representation[5], which is commonly used in information retrieval, the document i is represented by the q-dimensional vector

$$\mathbf{v}_i = (v_{i1}, \cdots, v_{iq}), \tag{5.1}$$

where v_{ij} is the $tf \times idf$ normalized term frequency[5] described as

$$v_{ij} = w_{ij} \log_2(\frac{n - n_j}{n_j}), \tag{5.2}$$

where n_j is the number of documents that include term j. The similarity between two documents $\mathbf{v}_i$ and $\mathbf{v}_j$ is defined by the cosine measure[5] as follows:

$$s(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i^T \mathbf{v}_j}{\| \mathbf{v}_i \| \| \mathbf{v}_j \|}. \tag{5.3}$$

5.3 Selecting Representative Documents

Let a document cluster $\mathcal{C}_k (k = 1, \cdots, c)$ is denoted as a set of the documents and described as

$$\mathcal{C}_k = \{\mathbf{v}_{k1}, \cdots, \mathbf{v}_{kn_k}\}, \tag{5.4}$$

where n_k is the number of documents in the cluster.

To inform users the contents of the document cluster $\mathcal{C}_k$, a representative document $\mathbf{r}_k$ is selected from $\mathbf{v}_{k1}, \cdots, \mathbf{v}_{kn_k}$ and its title is shown to users. The representative document is determined based on the sum of similarities to the other documents in the cluster:

$$E(\mathbf{v}_{ki}) = \sum_{j=1}^{n_k} s(\mathbf{v}_{ki}, \mathbf{v}_{kj}) \tag{5.5}$$

The document with the largest value of E is selected as the representative document $\mathbf{r}_k$.

5.4 Representative Documents of PAMI

In this section, we present the representatives extracted from real document images. These images consist of 97 papers and correspondences of *IEEE Transactions on Pattern Analysis and Machine Intelligence* (from Jan. 94 to Aug. 94). The first page of each paper is scanned and recognized. By a recognition method described in [36], the recognition rate was 99.0%. Here, most recognition faults are caused by segmentation faults. We used the word dictionary of *ispell* (103535 words) to verify extracted terms and the term that has no entry in the dictionary is not counted. As a result, the term loss ratio was 3.2%. We must note that the recognition rate and the term loss ratio are just estimated values based on the investigation of only 3 documents. We used the ARC method for clustering these documents.

Representative documents of PAMI are shown in Table 5.1. We also show a brief description about the topic of a representative and the number of documents

Table 5.1: Representative documents of PAMI

Topic	Documents	Representative
3-D Vision	19	Shape from focus (pp.824-830)
2-D Figure Analysis	14	Local versus Nonlocal Computation of Length of Digitized Curves (pp.726-733)
Character Recognition	9	Recognition of Handwritten Cursive Arabic Characters (pp. 664-672)
Classifier	7	Feature Preserving Clustering ··· (pp. 554-560)
Texture	6	Texture Segmentation Using ··· (pp. 130-149)
Morphology	6	Algorithms for the Decomposition of Gray-Scale Morphological Operations (pp.581-588)
Others	36	

considered to be relevant to each topic. It seems that the representatives reflect most important topics. However, we suppose that you can not judge our browsing system is good or bad only from this result. In the next section, we will establish the goodness measure of representative documents and evaluate our system quantitatively.

5.5 Performance Evaluation

The plain-text database we used was the CF database[37]. This database consists of 1239 medical documents, but the documents with abstracts are only 683. Since we used the abstracts to index documents, we rebuilt our database from 683 documents. We applied three clustering methods(the ARC method, Complete Link and Group Average) and random choosing to produce document clusters.

We defined the score of representatives as the average of the similarity between a document and the representative that is most similar to it.

$$score\_of\_representatives = \sum_{i=1}^{n} \max_{k=1,\cdots,c} s(\mathbf{v}_i, \mathbf{r}_k) \tag{5.6}$$

Fig. 5.3 shows the score of representatives of the database. We extracted 3-19 representatives using 3 clustering methods and random choosing. The score of random choosing was averaged over 50 trials. Naturally, the score of random choosing is the worst, and this is the lowerbound of the score. The score of the ARC method compares favorably with that of Complete Link, regardless of the number of representatives. The score of Group Average is almost equal to the other two clustering methods when the number of representatives are few, but as the number of representatives increases, the score gets worse. This result means the representatives produced by the ARC method are comparable to those by Complete Link and are better than those by Group Average.

Fig. 5.4 shows the computation time of Complete Link and the ARC method. In the eigenvector calculation of the ARC method, we used the Lanczos method[38], which is an efficient method for calculating several eigenvectors corresponding to the largest eigenvalues. The time complexity of the Lanczos method is $O(n^2)$[39]. This experiment was performed on a random sparse similarity matrix whose non-zero ratio is 10%, since the similarity matrix of documents is usually sparse[24]. The machine used here was Sun SS10(50Mhz). As a result, the ARC method was faster than Complete Link although it needs floating point calculations.

5.6 Experiment on OCR-generated Documents

In this section, we deal with OCR-generated documents that contain recognition faults and examine the effect of recognition faults on the tightness of document clusters. The experiment of document clustering needs a large number of texts, but it is very hard job to scan and recognize a large number of document images. Moreover, minute control of recognition rate by corrupting real images is very difficult. So, we employed an OCR simulation system, SimOCR[40], which substitutes a character for another character according to an OCR model. We also simulated the postprocessing facility that can recover a term with one fault, and terms with more than 2 faults

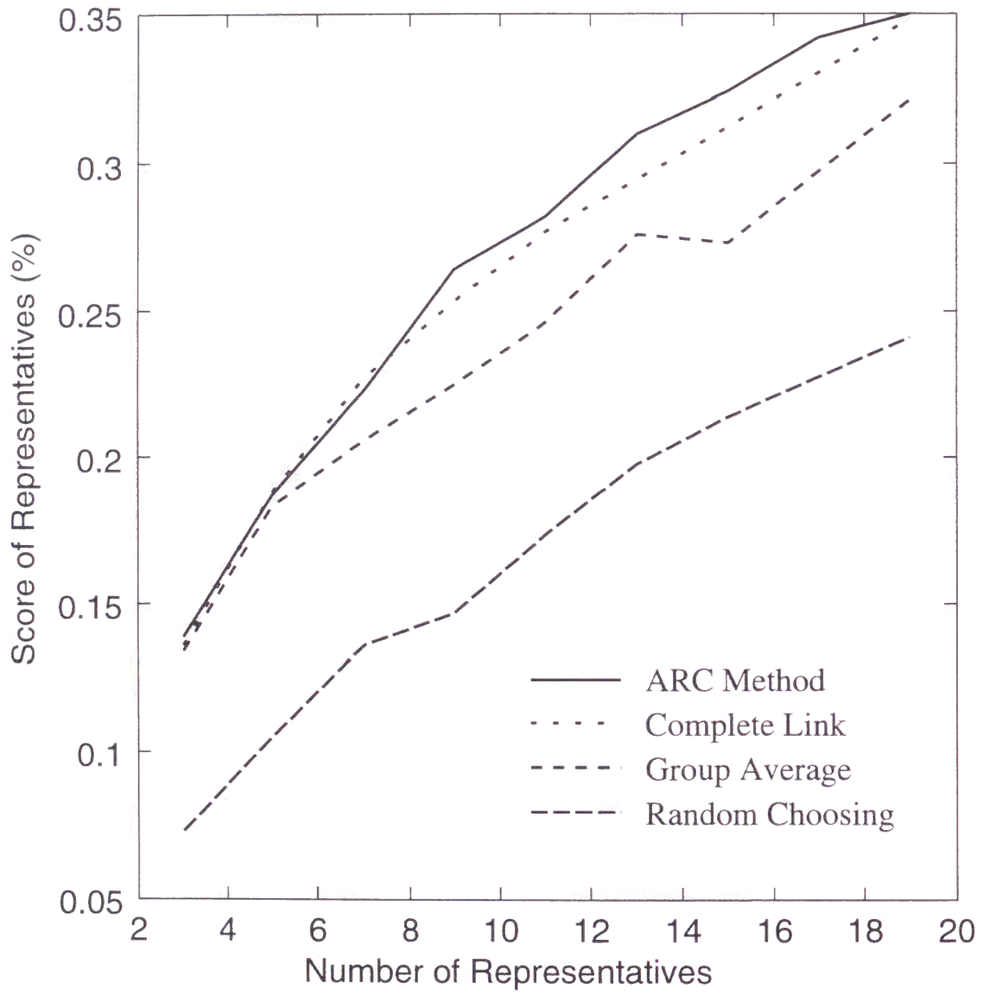


Figure 5.3: Score of representative documents of CF database

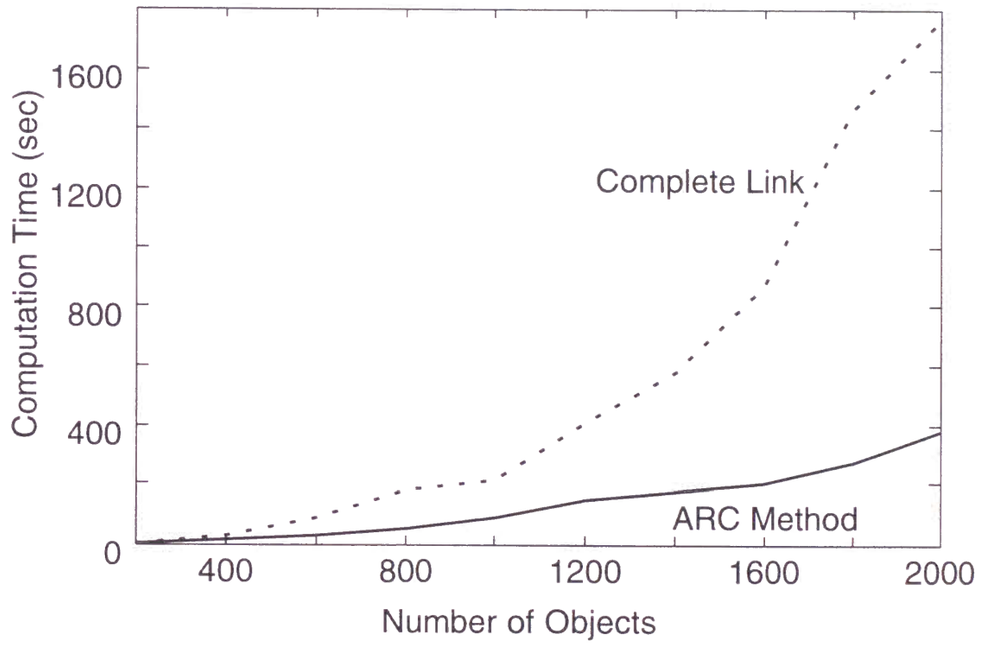


Figure 5.4: Computation Time of the ARC method and Complete Link

were discarded. We used the same 683 abstracts from CF database as an input to SimOCR. As a result, the term loss ratio was 22.7% at 85% recognition rate, 14.4% at 90%, 4.3% at 95%, and 0.5% at 99%, respectively.

Fig. 5.5 shows the scores of the ARC method and Complete Link at various term loss ratio. Clustering was performed with the similarities changed by term loss and the scores were calculated with original similarities. Here, the number of representatives was fixed on 15. To evaluate the effect of the term loss to the users' browsability, it is necessary to make clear the correspondence between the score and the browsability. To discuss this correspondence, we refer to another research about cluster-based browsing[15]. Since Group Average-based clustering is used and gives good browsability in this research, we assume that Group Average can produce tight clusters enough to be used for browsing. According to Fig. 5.3, the score of Group Average at 15 representatives is 0.273. In Fig. 5.5, the score of the ARC method is more than 0.273, when the term loss ratio is less than 24.4%. So, we conclude that the ARC method can make clusters usable for browsing up to 24.4% term loss ratio(84.4% recognition rate) at least.

5.7 Experiment on Term Clustering

Clustering terms can be performed in the similar way as clustering documents. The term j is represented by a n-dimensional vector of the occurence frequencies in the documents:

$$\mathbf{h}_j = (w_{1j}, \cdots, w_{nj}). \tag{5.7}$$

Then, the similarity between the two terms is obtained by the cosine measure.

$$s(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i^T \mathbf{h}_j}{\| \mathbf{h}_i \| \| \mathbf{h}_j \|}. \tag{5.8}$$

We extracted term clusters from a document database of the abstracts of 100 Japanese papers published by the information science department in Kyoto University in order to examine the robustness against trivial terms. The terms which are

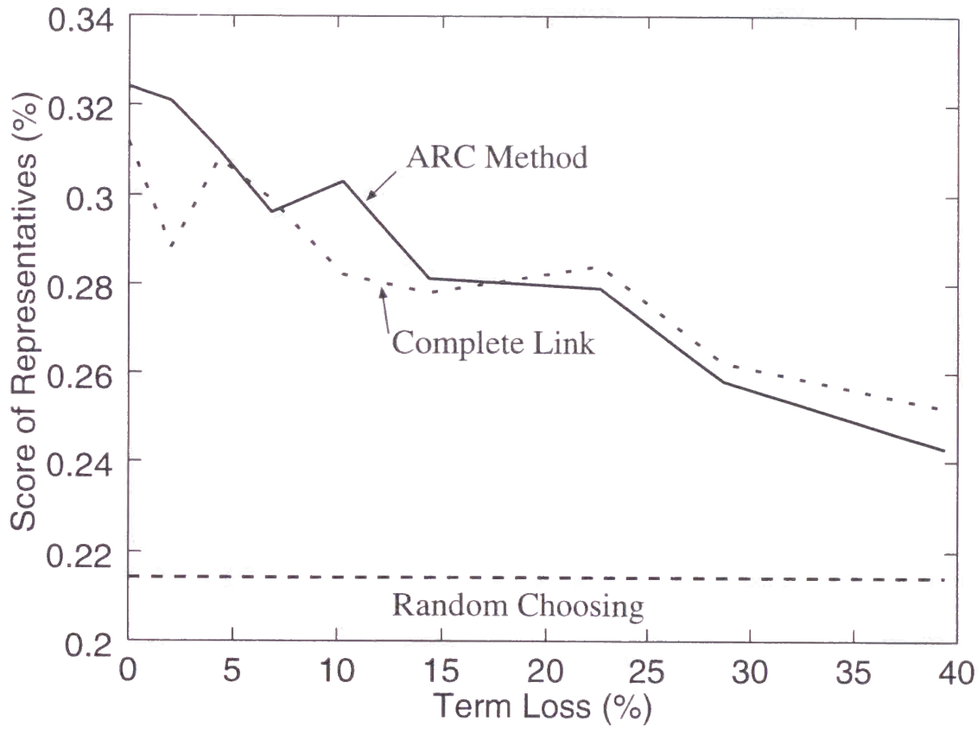


Figure 5.5: Score of representative documents at various term loss ratio

Table 5.2: The document database used in the term clustering experiment

Name	Thesis
The number of documents	100
The number of unique terms	2329
The number of terms	14473
The number of terms in each document	94∼234
The property of documents	Abstracts of scientific papers

contained in the dictionary (19653 Japanese terms) were extracted using the exhaustive text search algorithm of Senda. et. al[41] This database is called "Thesis" and the properties of this database are shown in Tab. 5.2.

The ARC method was performed and we obtained 6 clusters. We will show that the trivial terms are excluded from the clusters. To evaluate the degree that a term is trivial, we defined the "degree of importance" α_j as follows[5]:

$$\alpha_j = \max_{i=1,\cdots,n} w_{ij} \log_2 \frac{n - a_j}{a_j}, \tag{5.9}$$

where

$$a_j = \sum_{i=1}^{n} w_{ij}. \tag{5.10}$$

When the terms are concentrated on a small number of documents, α_i is large. For trivial terms, the occurence frequencies tend to be constant over all documents, so a trivial term has a small degree of importance.

In this experiment, we measured the averages of the degree of importance α_i over the following three term sets.

- Term set A: All the terms in extracted clusters.
- Term set B: The b terms which have the largest total occurence frequencies, where b is the number of terms extracted in the clusters.

Table 5.3: The most frequently occuring 17 terms in the database (The ones which are not extracted as clusters are underlined)

Terms	Processing	Parallel	Image	Research	System
Frequencies	238	160	139	132	131
Information	*Contents*	Recognition	Logic	Character	*Important*
125	107	105	103	102	89
Model	*Abstract*	*Computer*	Language	Development	*Object*
83	83	81	80	79	71

- Term set C: Randomly chosen b terms.

In Fig. 5.6, the averages of the degree of importance over the three sets against the number of clustered terms b are shown. The number of clustered terms varies due to the threshold η in Chap. 3. Since the degree of importance of the set A is larger than those of B and C, it is considered that the extracted clusters have the tendency to exclude trivial terms. Tab. 5.3 shows the most frequently occuring 17 terms. We underlined the terms that are not extracted as clusters. you can see that the trivial terms such as "important" and "contents" are excluded.

5.8 Summary

In this chapter, we applied the ARC method to document clustering and evaluated it as a document database browser. As a result, the clusters tighter than agglomerative methods could be obtained in less computation time, which is desirable for extracting representative documents. We also investigated the effect of character recognition faults on document clustering and found our method can make tight document clusters usable for browsing up to 24.4% term loss ratio. In the future work, we will combine our system with an IR system and evaluate the user supporting performance of our system.

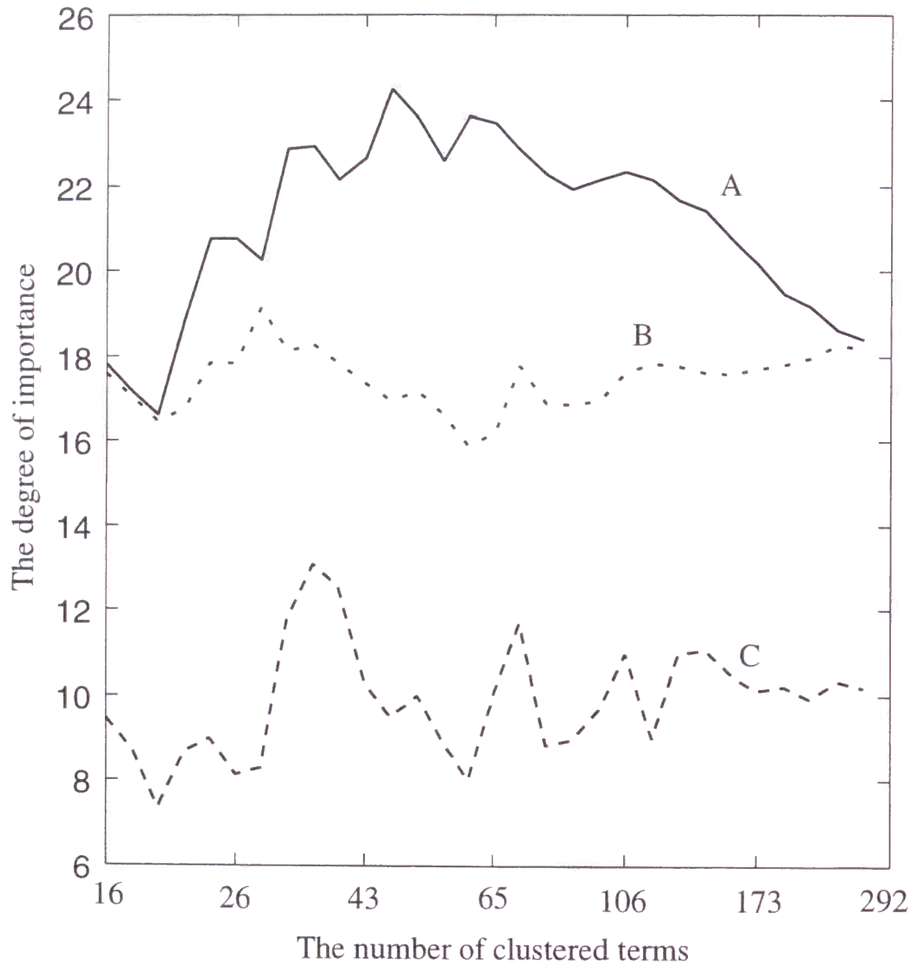


Figure 5.6: The average of the degree of importance over term sets A, B and C against the number of clustered terms)

Chapter 6

Application: Generating Prototypes from Training Samples

6.1 Introduction

Pattern recognition is the task to assign a label of a priori defined classes to an object. An unlabeled sample $\mathbf{x} \in \Re^d$ is classified to the class $\Gamma_\ell(\ell = 1, \cdots, q)$ according to various rules. The rules are inferred from the examplar samples that are already labelled. These samples are called "training samples". The training samples of class Γ_ℓ are denoted as $\mathbf{t}_{\ell i}(i = 1, \cdots, n)$.

The nearest neighbor method[6] is the most fundamental pattern recognition method. When the unlabeled sample $\mathbf{x}$ is given, the nearest training sample is searched, and $\mathbf{x}$ is classified to the class which the nearest training sample belongs to. Because of its simplicity, the nearest neighbor method is used in many tasks[42, 43], and there are a lot of theoretical works about it[44, 45].

But, the computational cost of the nearest neighbor is large in comparison with other classifiers, because the distance to every training sample has to be computed. To reduce the computational cost, the reduction of the training samples is required.

The reduced training samples are called "prototypes"[46]. The prototypes should be generated so that the loss of the classification accuracy is minimized.

Clustering methods are often used for this purpose[47, 46, 48]. The training samples are partitioned into several clusters and the reduced training set (i.e. the prototype set) is obtained as the cluster centers.

The points which are classified to the class Γ_ℓ by the nearest neighbor method form a region $\mathcal{R}_\ell$:

$$\mathcal{R}_\ell = \bigcup_{i=1}^{n} \mathcal{T}_{\ell i}, \tag{6.1}$$

where

$$\mathcal{T}_{\ell i} = \{\mathbf{x} \mid \| \mathbf{x} - \mathbf{t}_{\ell i} \| < \| \mathbf{x} - \mathbf{t}_{kj} \|, \ (\forall k \neq \ell, \forall j \neq i)\}. \tag{6.2}$$

We call this region "the class region". The whole feature space is partitioned by the class regions thoroughly. Since the classification result of the nearest neighbor method depends solely on the partition, the prototypes should preserve the partition in order not to reduce the loss of the classification accuracy.

In the prototype generation, the most frequently used method is the C-Means method[16, 46, 47]. But, due to the weakness against noise, the C-Means method tends to change the partition largely due to a few noise training samples, which are characterized as the training samples whose distances to the others are large. A noise training sample forms a isolated region from the region of the majority of the training samples, and the regions of other classes lie between the two regions (Fig. 6.1).

Since the C-Means classifies all the samples to clusters, the noise training sample is classified into a certain cluster. By the inclusion of the noise training sample, the center of the cluster is biased toward the noise training sample, because the center is obtained as the arithmetic mean of the samples in the cluster. The bias of the center is reflected as the bias of the prototype and, as a result, the class region grows toward the noise sample and invades the regions of the other classes. The invasion changes the partition substantially and leads to the loss of the classification accuracy.

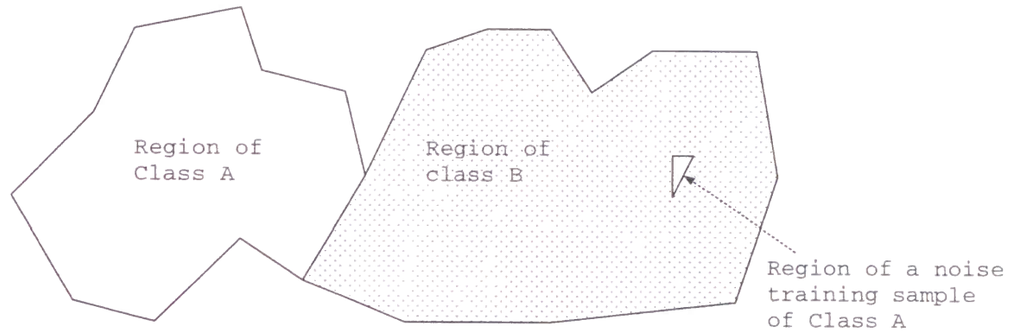


Figure 6.1: The class regions with a noise training sample

When there are noise training samples, the clustering method that has the robustness against noise is needed to avoid the loss of the classification accuracy. We have already shown that the ARC method has the high robustness against noise in Chap. 3, and so the prototypes generated by the ARC method might have less loss of the classification accuracy than those by C-Means.

In the experiments of 3D object recognition[17] and Hiragana recognition[19, 49, 18], we examined the classification accuracy of the nearest neighbor method with the prototypes generated by the ARC method and the C-Means. As a result, the ARC method achieved the higher classification accuracy with the same number of the prototypes. This result shows that robustness against noise is needed to generate the prototypes with the small loss of the classification accuracy.

The rest of this chapter is organized as follows: In Sec. 6.2, the prior works on pattern recognition are briefly reviewed. In Sec. 6.3, we will explain how to generate the prototypes by clustering. In Sec. 6.4, the 3D object recognition experiment is performed to evaluate the prototypes. In Sec. 6.5, the Hiranaga recognition experiment is also carried out. Sec. 6.6 describes the summary.

6.2 Prior Works on Pattern Recognition

In this section, a brief review of pattern recognition is made from our own point of view. In classifying pattern recognition methods, there are many ways: parametric vs nonparametric, nonlinear vs linear and so on. But, one of the most important distinctions is that the classifier is prediction-based or regression-based.

Prediction-based methods use the distances to classes in classification. The distance to a class is derived as follows (Fig. 6.2): First, a class Γ_ℓ is described with a set of points in the feature space called a predictor set $\mathcal{P}_\ell$[50]. In discrete cases, the predictor set $\mathcal{P}_\ell$ consists of the training samples $\mathbf{t}_{\ell 1}, \cdots, \mathbf{t}_{\ell n}$, or the prototype samples derived from the training samples. In continuous cases, the predictor set is described by a function of the training samples:

$$\mathcal{P}_\ell = \{\mathbf{x} | \mathbf{x} = \mathbf{f}_\ell(\mathbf{t}_{\ell 1}, \cdots, \mathbf{t}_{\ell n})\}. \tag{6.3}$$

This function is called a predictor function. Then, when an unlabeled sample $\mathbf{x}$ is given to be classified, the distance to $\mathcal{P}_\ell$ is obtained based on a user-defined distance measure $M(\mathbf{x}, \mathcal{P}_\ell)$. A frequently used distance measure is the nearest neighbor metric:

$$M(\mathbf{x}, \mathcal{P}_\ell) = \min_{\mathbf{p} \in \mathcal{P}_\ell} \| \mathbf{x} - \mathbf{p} \|^2 . \tag{6.4}$$

Then, $\mathbf{x}$ is classified to the class with smallest $M(\mathbf{x}, \mathcal{P}_\ell)$.

In discrete cases, the training is performed by choosing the prototype samples. In continuous cases, the training is done by determining the parameters in the predictor function $\mathbf{f}_\ell$. Usually, the training is formulated as an optimization problem such that a specified criterion is optimized.

The prediction-based methods include Nearest neighbor[6], LVQ (learning vector quantization)[51], Gaussian mixture[6], subspace methods[25], and several methods using splines as the predictor function[17, 52].

In regression-based methods, the classification is performed using a regression function $g_\ell(\mathbf{x}, \boldsymbol{\theta}), \mathbf{x} \in \Re^d$, where $\boldsymbol{\theta}$ is a vector of parameters (Fig. 6.3). The parameters are trained so that $g_\ell(\mathbf{x}, \boldsymbol{\theta})$ indicates 1 and 0 when $\mathbf{x}$ belongs to class Γ_ℓ

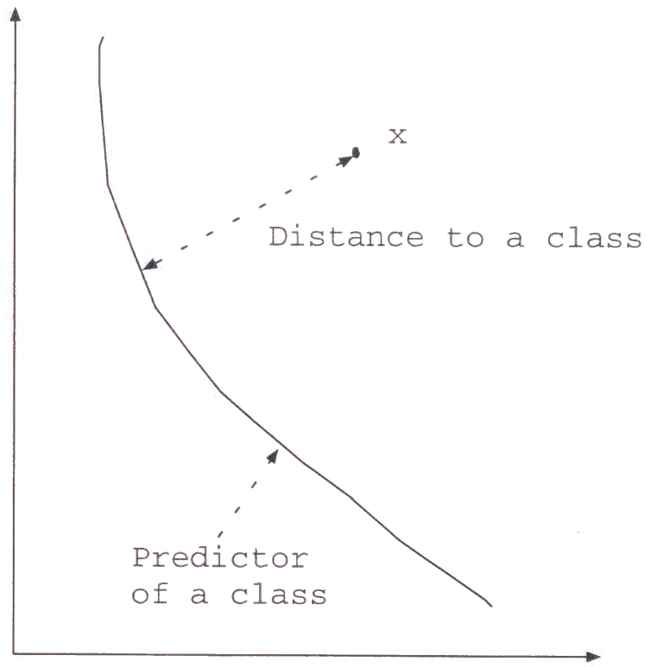


Figure 6.2: Prediction-based Methods

and other classes, respectively. The optimal $\boldsymbol{\theta}$ can be obtained as a solution of the following simultanous equations.

$$g_\ell(\mathbf{t}_{\ell i}, \boldsymbol{\theta}) = 1 \quad (i = 1, \cdots, n) \tag{6.5}$$

$$g_\ell(\mathbf{t}_{ki}, \boldsymbol{\theta}) = 0 \quad (i = 1, \cdots, n, k \neq \ell), \tag{6.6}$$

where $\mathbf{t}_{\ell i}$ and $\mathbf{t}_{ki}$ are the training samples of class Γ_ℓ and Γ_k, respectively. These equations are often nonlinear, so a nonlinear optimization method is required to obtain a solution. In classification, the unlabeled sample $\mathbf{x}$ is substituted into $g_\ell(\mathbf{x}, \boldsymbol{\theta})$, and the sample is classified to the class with the largest value.

Most of the neural networks classifiers belong to the regression methods. Two major ones are multilayer perceptrons[53] and radial basis function (RBF) networks[54]. Also, any regression method can be applied to pattern recognition in the same manner.

6.2.1 Nearest Neighbor

The nearest neighbor method is a prediction-based method, where the predictor set is the set of training samples,

$$\mathcal{P}_\ell = \{\mathbf{t}_{\ell 1}, \cdots, \mathbf{t}_{\ell n}\}, \tag{6.7}$$

and the distance measure to the predictor set is the nearest neighbor distance:

$$M(\mathbf{x}, \mathcal{P}_\ell) = \min_{\mathbf{p} \in \mathcal{P}_\ell} \| \mathbf{x} - \mathbf{p} \|^2 . \tag{6.8}$$

There is a extended version called K-nearest neighbor[42, 43], where an unlabeled sample is classified to the class which has the largest number of samples in the k-nearest samples.

The advantage of the nearest neighbor method is its simplicity. It is easy to implement and to analyze. But, it is known that the classification accuracy of this method is not high, especially when the number of samples is small[55]. The situations that the nearest neighbor method can be used are limited to very easy pattern recognition problems where the classes have almost no overlaps.

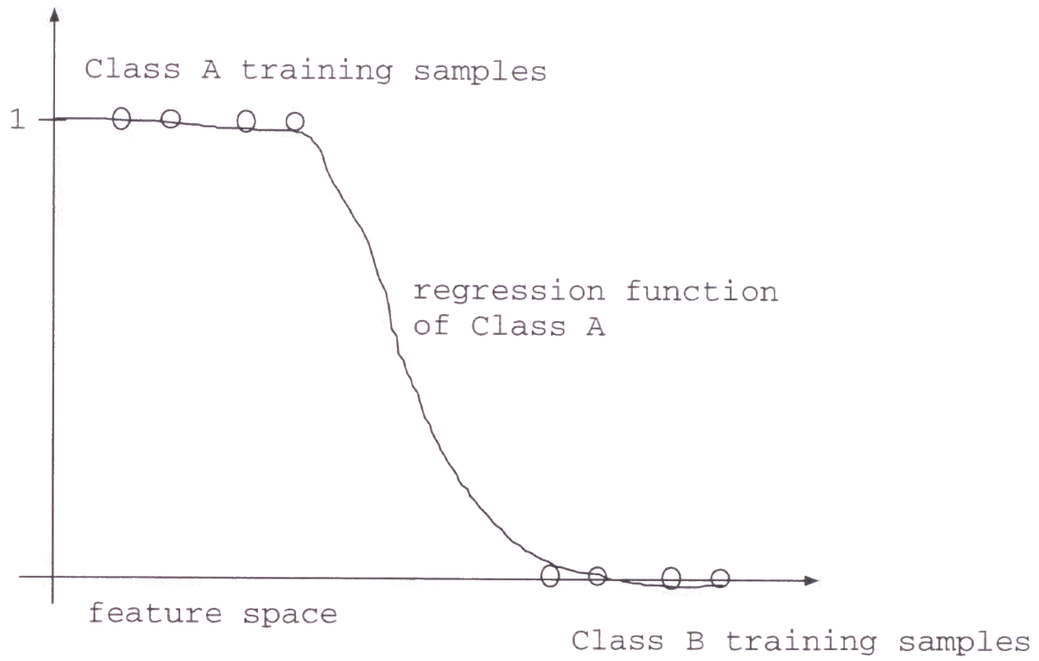


Figure 6.3: Regression-based method in the case of 2 classes

6.2.2 LVQ

The learning vector quantization (LVQ) is an expanded form of the nearest neighbor method. The predictor set consists of the prototype samples:

$$\mathcal{P}_\ell = \{\mathbf{q}_{\ell 1}, \cdots, \mathbf{q}_{\ell m}\}, \tag{6.9}$$

and the distance measure to $\mathcal{P}_\ell$ is the nearest neighbor metric.

The prototype samples are determined so that each of the training samples is classified to its own class. Although many improved forms are proposed, the most basic algorithm is described as follows[53]:

- The prototype samples are initialized randomly.
- For each of the training samples $\mathbf{t}_{\ell i}(\ell = 1, \cdots, c, i = 1 \cdots, n)$, the following procedure is repeated.
  - Let the nearest prototype sample from $\mathbf{t}_{\ell i}$ be $\mathbf{q}_{rj}$.
  - if $r = \ell$, the prototype sample $\mathbf{q}_{rj}$ is moved to

$$\mathbf{q}_{rj} + \alpha(\mathbf{x} - \mathbf{q}_{rj}). \tag{6.10}$$

  - if $r \neq \ell$, the prototype sample $\mathbf{q}_{rj}$ is moved to

$$\mathbf{q}_{rj} - \alpha(\mathbf{x} - \mathbf{q}_{rj}). \tag{6.11}$$

It is desirable for the learning constant α to decrease monotonically with the number of iterations. For example, α may initially be about 0.1, or smaller, and the decrease linearly with the number of iterations. After several passes through the input data, the prototype samples would typically converge, and the training is complete.

6.2.3 Gaussian Mixture

The Gaussian mixture method[6, 56] is also a prediction-based method, where the class conditional distribution is assumed to be the mixture of sevaral Gaussian distributions[6]. The predictor set is the prototype samples:

$$\mathcal{P}_\ell = \{\mathbf{q}_{\ell 1}, \cdots, \mathbf{q}_{\ell m}\}, \tag{6.12}$$

and the distance measure is

$$M(\mathbf{x}, \mathcal{P}_\ell) = -\log(\sum_{i=1}^{m} \alpha_{\ell i} \exp(-\frac{\| \mathbf{x} - \mathbf{q}_{\ell i} \|^2}{\sigma_{\ell i}^2})), \tag{6.13}$$

where

$$\sum_{i=1}^{m} \alpha_{\ell i} = 1. \tag{6.14}$$

The training is performed so that

$$\sum_{i=1}^{n} M(\mathbf{t}_{\ell i}, \mathcal{P}_\ell) \tag{6.15}$$

is minimized. Here, there are three kinds of parameters, $\alpha_{\ell i}$,$\mathbf{q}_{\ell i}$ and $\sigma_{\ell i}$, so the optimization of these parameters is extremely difficult. Therefore, the application of the Gaussian mixture to real-world large-scale problems is not realistic.

6.2.4 Multilayer Perceptron

The multilayer perceptron is a regression-based method where the regression function is composed of sigmoid functions[53, 57]. The network structure of standard three-layered perceptron is shown in Fig. 6.4. Each node performs as an activation function:

$$z(\sum_{i=1}^{p} w_i x_i), \tag{6.16}$$

where $x_1, \cdots, x_p$ are input variables, $w_1, \cdots, w_p$ are parameters, and z is a sigmoid function such as

$$z(x) = \frac{1}{1 + \exp(-x)}. \tag{6.17}$$

Therefore, the regression function of the multilayer perceptron is as follows:

$$g_\ell(\mathbf{x}) = z(\sum_{j=0}^{m} w_{\ell j}^{(2)} z(\sum_{i=0}^{p} w_{ji}^{(1)} x_i)), \tag{6.18}$$

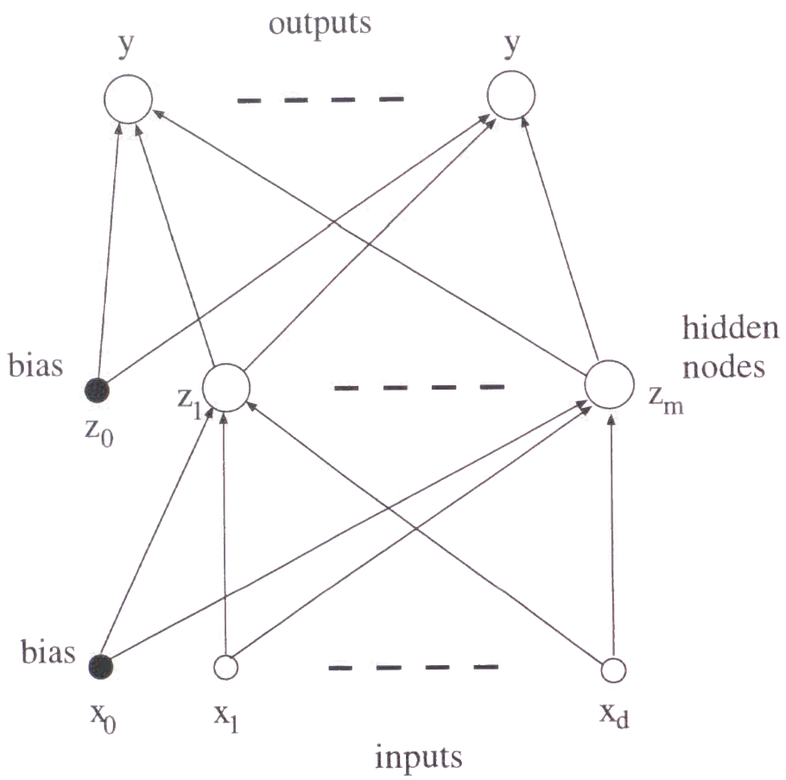


Figure 6.4: An example of a multilayer perceptron having three layers. The bias parameters are shown as weights from extra inputs having a fixed value 1

where $w_{ji}^{(1)}$ are the weight parameters in the first layer, $w_{\ell j}^{(2)}$ are those in the second layer, and m is the number of hidden nodes. Since the regression function is highly nonlinear, local minima problem is very serious in the multilayer perceptron. There is a parallel optimization algorithm called "back propagation"[6]. However, this algorithm cannot avoid local minima, and much worse, it is not guaranteed to converge to local minima[53].

6.2.5 RBF Networks

The RBF network[6, 54, 58] is a regression-based method where the regression function is summarized by the following general form:

$$g_\ell(\mathbf{x}) = \sum_{i=1}^{m} w_{\ell i}\Phi((\mathbf{x}-\mathbf{y}_i)^T(\mathbf{x}-\mathbf{y}_i)), \tag{6.19}$$

where $\Phi((\mathbf{x}-\mathbf{y}_i)^T(\mathbf{x}-\mathbf{y}_i))$ denotes the kernel function whose central point is $\mathbf{y}_i \in \Re^p$, and $w_{\ell i}$ denotes the weight value assigned to the kernel function. Normally, Gaussian function is chosen as the kernel function, that is,

$$\Phi(z) = \exp(\frac{-z}{\sigma^2}), \tag{6.20}$$

where σ is a parameter that determines the width of Gaussian. The RBF network consists of m kernel function nodes and one sum-up node as shown in Fig. 6.5.

When the width is fixed to a certain value and the central points are chosen from training samples, that is,

$$\mathbf{y}_i \in \{\mathbf{t}_{\ell j} | \ell = 1, \cdots, c, j = 1, \cdots, n\}, \tag{6.21}$$

the weight parameters can be obtained analytically. Let the m-dimensional vector of weight parameters be $\mathbf{w}_\ell$. This vector can be obtained as the solution of the following linear simultanous equations:

$$A\mathbf{w}_\ell = \mathbf{b}_\ell, \tag{6.22}$$

where A is a $m \times m$ matrix whose (i,j) element is

$$a_{ij} = \Phi((\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{y}_i - \mathbf{y}_j)), \tag{6.23}$$

and $\mathbf{b}_\ell$ is a m dimensional vector such that

$$b_{ki} = \begin{cases} 1 & \mathbf{y}_i \in \Gamma_\ell \\ 0 & \mathbf{y}_i \notin \Gamma_\ell \end{cases}. \tag{6.24}$$

However, if the central positions and the width are assumed to be variables, the nonlinear optimization is needed for training[59].

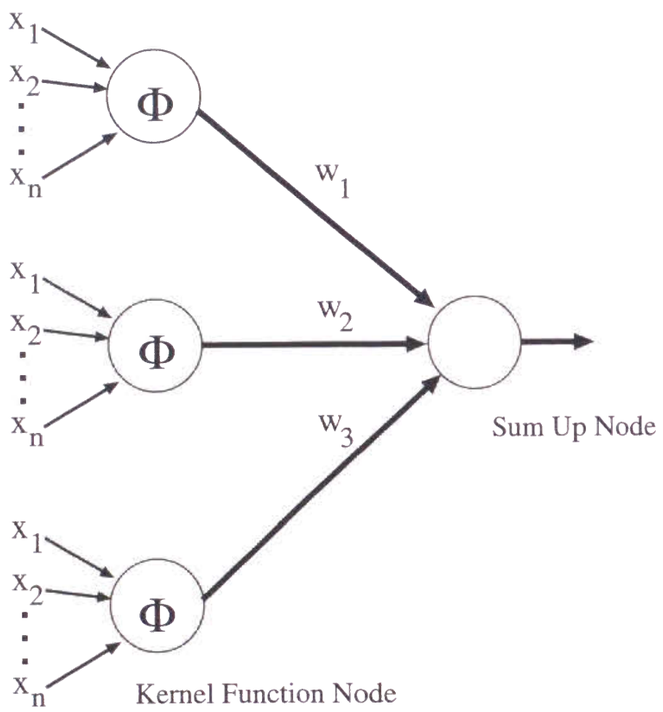


Figure 6.5: Network structure of RBF network

6.3 Generating Prototypes by Clustering

In the generation of the prototypes, the m prototypes $\mathbf{p}_{\ell 1}, \cdots, \mathbf{p}_{\ell m}$ are generated from the training sample set of the class Γ_ℓ:

$$\mathcal{T}_\ell = \{\mathbf{t}_{\ell 1}, \cdots, \mathbf{t}_{\ell n}\}. \tag{6.25}$$

When the clusters $\mathcal{C}_1, \cdots, \mathcal{C}_m$ is obtained from $\mathcal{T}_\ell$, one prototype is generated from a cluster as the mean of all the samples in the cluster:

$$\mathbf{p}_{\ell i} = \frac{1}{n_i} \sum_{\mathbf{t} \in \mathcal{C}_i} \mathbf{t}, \tag{6.26}$$

where n_i is the number of elements of $\mathcal{C}_i$.

6.4 3D Object Recognition Experiment

In this section, the performance of the nearest neighbor method with the prototypes generated by the C-Means method and the ARC method is examined in the 3D object recognition experiment. The 3D object recognition is selected here as it is a typical and practical pattern recognition problem. Eight objects shown in Fig.6.6 are used. The objects are rotated to random directions around the center of gravity, and projected to 2D images. As a result, 80 images are obtained for each object (Fig. 6.9 and Fig. 6.10). A half of those images are used for training and the others are used for testing.

6.4.1 Derivative of Gaussian Filter

In feature extraction, we used the derivative of Gaussian (DOG) filters[60], because they are considered as the good approximation of the feature extration filters in the human fovea[60]. In this method, the Gaussian filter,

$$G(x, y) = \exp(-\frac{x^2 + y^2}{\rho^2}), \tag{6.27}$$

is differentiated to be directional filters:

Figure 6.6: 3D objects used for the experiment

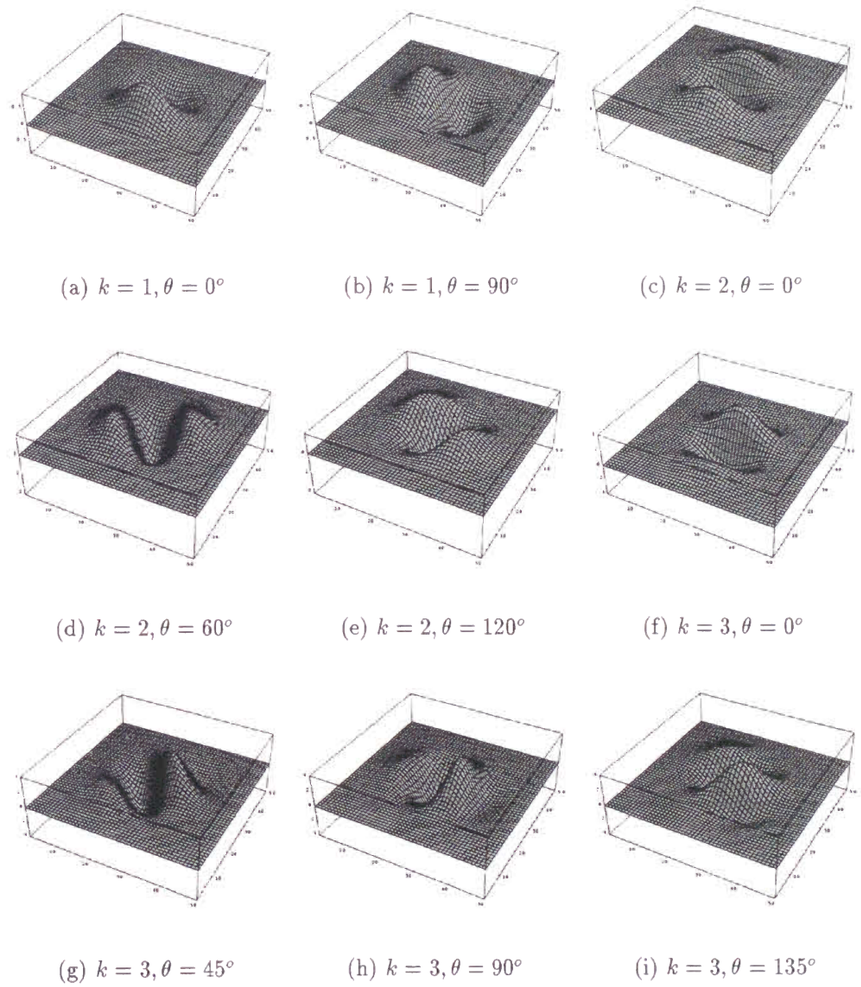
$$G_k(x,y) = \frac{d^k}{dx^k} G(x,y). \tag{6.28}$$

The DOG filters are rotated to be selective for particular directions, where the angle of rotation is denoted as θ. In this experiment, the nine filters shown in Fig. 6.7 were used in the 12x12 grids on the image. Therefore, the dimensionality was 12x12x9=1296 here.

6.4.2 Experimental Result

The error rates of the nearest neighbor method based on the prototypes generated by the ARC method and the C-Means method are shown in Fig. 6.8. The number of prototypes was changed from 2 to 10. In the ARC method, η was set to 0.25 and σ was set to 2300. In C-Means, ten trials were made with different initial cluster centers, and the one trial which achieved the smallest error is adopted.

When the number of prototypes was small, the error rate of the C-Means is smaller than that of the ARC method. It shows that the prototypes by the ARC method are concentrated in a small part of the class region and do not cover the



(a) $k = 1, \theta = 0^o$ (b) $k = 1, \theta = 90^o$ (c) $k = 2, \theta = 0^o$

(d) $k = 2, \theta = 60^o$ (e) $k = 2, \theta = 120^o$ (f) $k = 3, \theta = 0^o$

(g) $k = 3, \theta = 45^o$ (h) $k = 3, \theta = 90^o$ (i) $k = 3, \theta = 135^o$

Figure 6.7: Nine derivative of Gaussian filters used in the experiment

whole region, because the number of the prototypes is too small. But, as the number of the prototypes increases, the ARC method outperforms the C-Means method. The result shows that the robustness against noise works to improve the classification accuracy.

6.5 Hiragana Recognition Experiment

Here, we deal with 71 Hiraganas (Fig. 6.12), which is a small subset of Kanji characters, Character images are derived from the ETL9B database, which contains 200 images for each character. We used randomly chosen 160 characters for training and the rest 40 characters are used for testing.

6.5.1 Contour Direction Histgram Feature

In the feature extraction method called "contour direction histgram"(CDH) [18, 61], the character image is divided into 7x7 blocks first. In each block, the contour is traced and digitized into four directions (0,45,90,135 degrees), and the direction histgram is computed as shown in Fig. 6.13. As a result, characters are represented in a 196 (7x7x4) dimensional feature space.

6.5.2 Experimental Result

The error rates of the nearest neighbor method based on the prototypes generated by the ARC method and the C-Means method are shown in Fig. 6.14. The parameters of the ARC method were set as follows: $\eta = 0.4, \sigma = 20$, and we made ten trials in C-Means. We had a similar result also in Hiragana recognition experiments: The error rate of the ARC method was significantly smaller than that of C-Means method. The results show that the advantage of our method is not limited to particular problems, and it is suggested that the same result would appear in other problems.

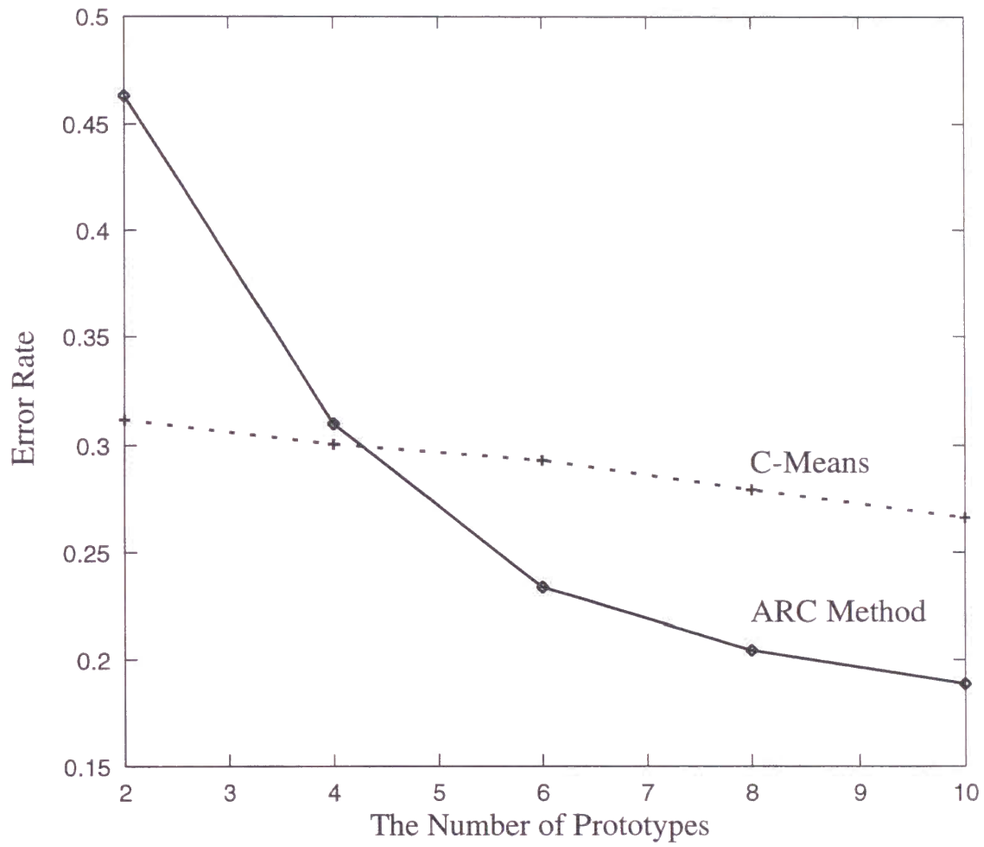


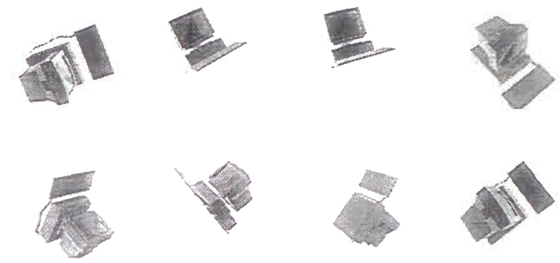
Figure 6.8: Error rates of the nearest neighbor classifier based on the prototypes generated by the ARC method and the C-Means in 3D object recognition

(a) al

(b) ant

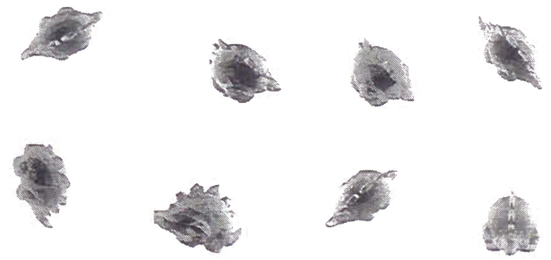
(c) beethovan

Figure 6.9: Examples of Rotated Objects (1)



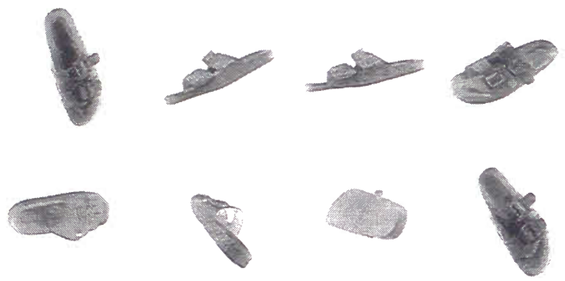
(a) computer

(b) cow

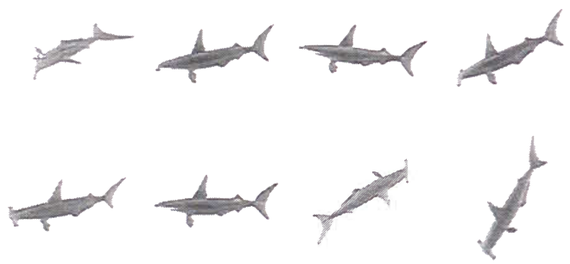


(c) dinosaur

Figure 6.10: Examples of Rotated Objects (2)



(a) sandal



(b) shark

Figure 6.11: Examples of Rotated Objects (3)

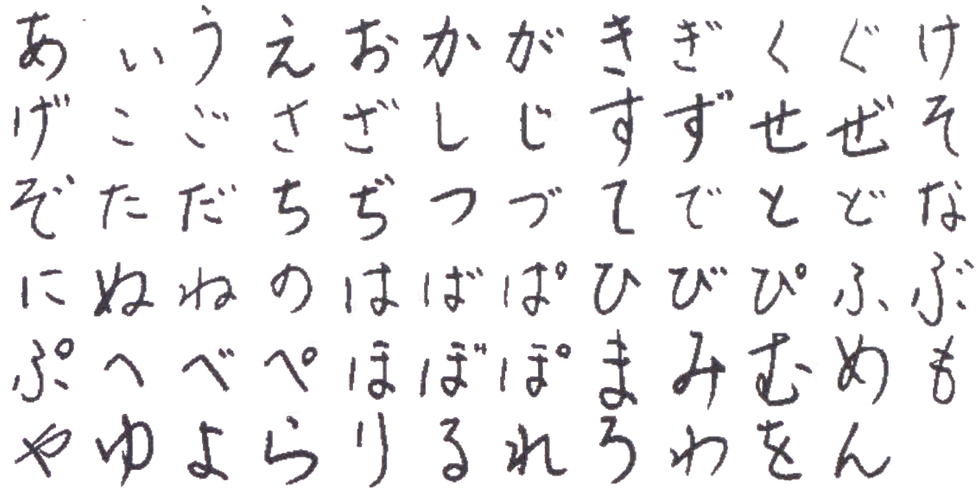


Figure 6.12: Samples of Hiragana (71 characters)

6.6 Summary

In this chapter, we applied the ARC method to generate the prototypes for the nearest neighbor method. We compared the goodness of the prototypes with those generated by the C-Means method with regard to the classification accuracy in the experiments of the 3D object recognition and the Hiragana recognition. As a result, it is shown that the robustness against noise works to improve the prototypes so that the error rate gets smaller.

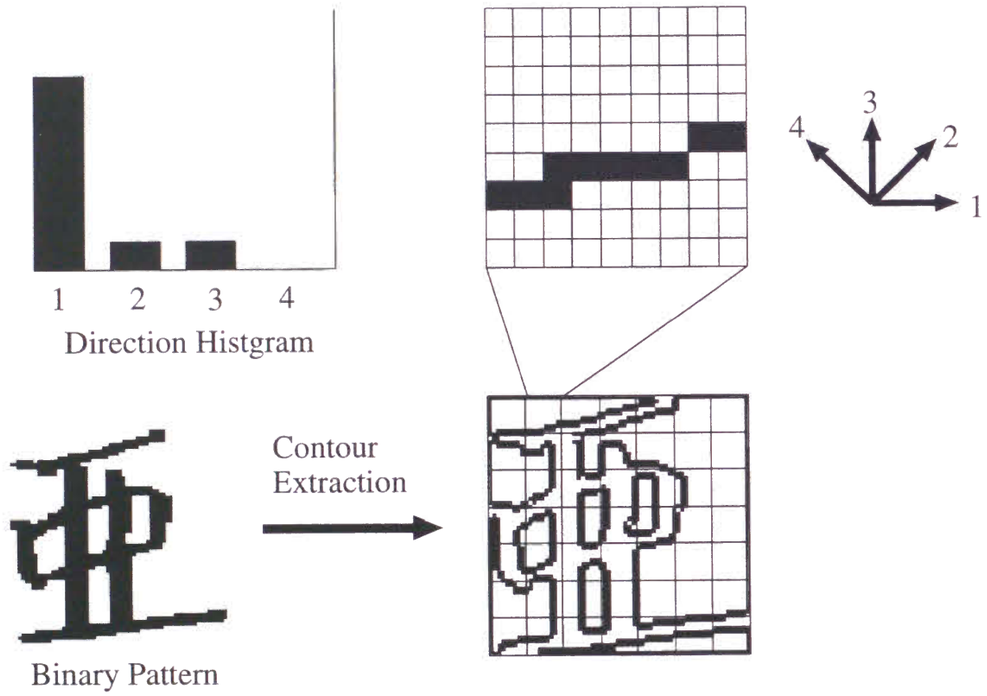


Figure 6.13: Contour Direction Histgram

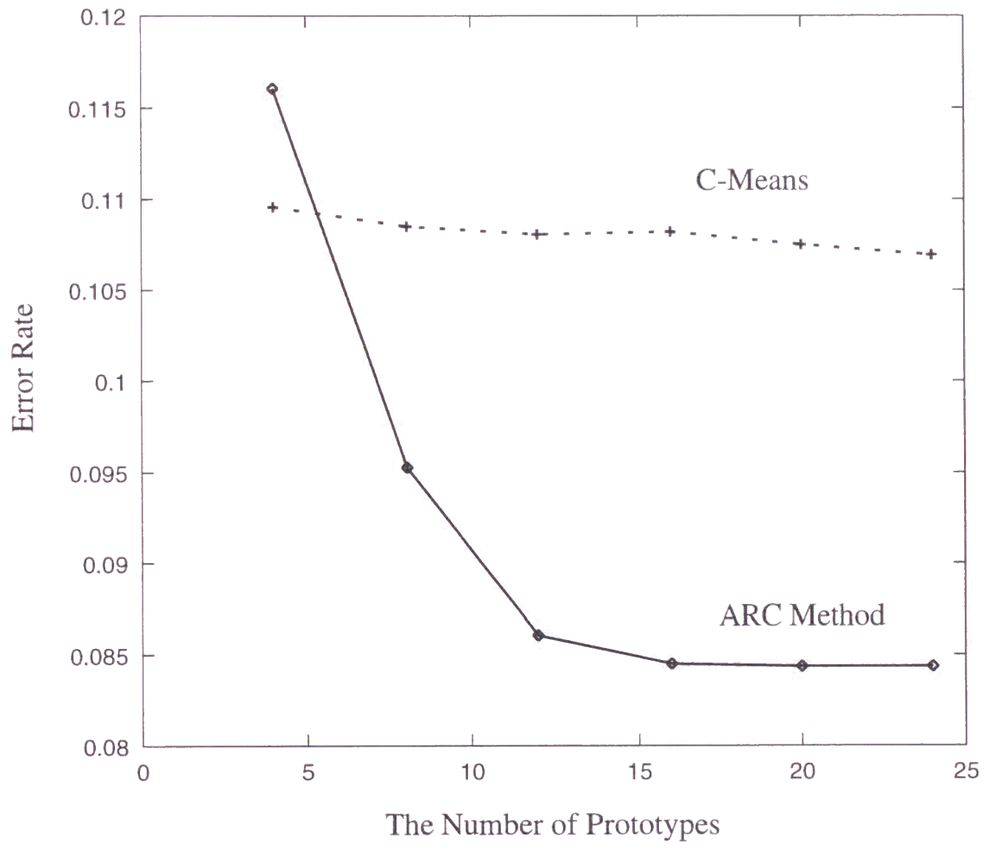


Figure 6.14: Error rates of the nearest neighbor classifier based on the prototypes generated by the ARC method and the C-Means in Hiragana recognition

Chapter 7

Conclusion

In this thesis, we proposed the clustering method that has the two important properties: the robustness against noise and the analytical computability. The name of this method is the "analytically-computable robust clustering method (ARC method)".

The ARC method is derived as the extension of the cone cluster extraction method, which is the extractive clustering method based on the cone distance. The cone cluster extraction has the advantage that the clusters can be obtained by the analytical computations, but has the drawback that it can only extract the cone-shaped clusters of the fixed size. Since the clusters are usually considered to be spherical, we added the preprocessing called "inner product scaling" which converts the spherical clusters into the cone-shaped clusters.

We have presented the three applications of the ARC method. First, the ARC method was applied to the line extraction from the images. In the line extraction by the clustering, the lines are extracted as the clusters of line segments, where the similarity is defined between the pairs of line segments so that the two segments aligned in line have a large similarity. The line segments which do not belong to any line are considered to be noises in the clustering. So, the robustness against noise is needed for this task. As a result of the experiments, it is found that, with regard to the frequency of the misextraction of the lines, the ARC method was superior to the conventional threshold-based methods.

Second, the ARC method was applied to the document clustering. In the document database browser, similar documents are clustered and a representative document of each cluster is shown to the user. The user can get the whole view of the database without examining documents one by one. In a document database, there are many documents whose contents are not similar to any document. Since such documents are considered as noise documents in the clustering, the ARC method is useful also in document clustering. As a result of the experiments, the ARC method extracted more informative representative documents than those of the conventional agglomerative methods.

Third, the ARC method was applied to the prototype generation for the nearest neighbor method. To reduce the computational cost of the nearest neighbor method, the reduction of the training samples are required. The training samples are partitioned into several clusters and the reduced training set (i.e. prototypes) is obtained as the cluster centers. The most frequently used method for the prototype generation is the C-Means method, but it is easily affected by the noise training samples. As a result of the experiments, we have shown that the ARC method can generate better prototypes than the C-Means with respect to the classification accuracy of the nearest neighbor method.

So far, the classification algorithms have developed from linear ones to nonlinear ones in order to achieve the high classification performance[6]. In course of the development, the analytical computability has been lost, and, as a result, the classification algorithms became very difficult to use. In this thesis, we proposed the analytically computable algorithm for the robust clustering. In the future works, we would like to develop the classification algorithms which have both of the analytical computability and the high performance.

Bibliography

[1] A. K. Jain and R. C. Dubes: "Algorithms for Clustering Data", Prentice Hall (1988).

[2] C. Carpineto and G. Romano: "Galois: An order-theoretic approach to conceptual clustering", Proceedings of the Tenth International Conference on Machine Learning, pp. 33–40 (1993).

[3] J. Segen: "Graph clustering and model learning by data compression", Proceedings of the Seventh International Conference on Machine Learning, pp. 93–98 (1990).

[4] N. Ueda and R. Nakano: "A competitive & selective learning method for designing vector quantizers", 1993 IEEE Int. Conf. Neural Netw., pp. 1444–1449 (1993).

[5] W. B. Frakes and R. Baeza-Yates Eds.: "Information Retrieval: Data Structures & Algorithms", Prentice Hall (1992).

[6] C. M. Bishop: "Neural Networks for Pattern Recognition", Oxford University Press (1995).

[7] M. M. Deza and M. Laurent: "Geometry of Cuts and Metrics", Springer (1997).

[8] M. M. Trivedi and J. C. Bezdek: "Low-level segmentation of aerial images with fuzzy clustering", IEEE Trans. Syst. Man Cybern., **SMC-16**, 4, pp. 589–598 (1986).

[9] J. Kim, R. Krishnapuram and R. Dave: "Application of the least trimmed squares technique to prototype-based clustering", Pattern Recognition Letters, **17**, 6, pp. 633–641 (1996).

[10] P. Meer, D. Mintz, A. Rosenfeld and D. Y. Kim: "Robust regression methods for computer vision: A review", Int. J. Comput. Vision, **6**, 1, pp. 59–70 (1991).

[11] R. N. Dave: "Characterization and detection of noise in clustering", Pattern Recognition Letters, **12**, 11, pp. 657–664 (1991).

[12] H. Frigui and R. Krishnapuram: "A robust clustering algorithm based on competitive agglomeration and soft rejection of outliers", Proc. CVPR '96, pp. 550–555 (1996).

[13] J.-M. Jolion, P. Meer and S. Bataouche: "Robust clustering with applications in computer vision", IEEE Trans. Patt. Anal. Mach. Intell., **13**, 8, pp. 791–801 (1991).

[14] I. Borg and P. Groenen: "Modern Multidimensional Scaling", Springer-Verlag (1997).

[15] D. R. Cutting, D. R. Karger, J. O. Pedersen and J. W. Tukey: "Scatter/gather: A cluster-based approach to browsing large document collections", SIGIR '92, pp. 318–329 (1992).

[16] J. C. Bezdek: "A convergence theorem for the fuzzy isodata clustering algorithms", IEEE Trans. Patt. Anal. Mach. Intell., **2**, 1, pp. 1–8 (1980).

[17] S. K. Nayar and T. Poggio: "Early Visual Learning", Oxford University Press (1996).

[18] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake: "Modified quadratic discriminant functions and the application to chinese character recognition", IEEE Trans. Patt. Anal. Mach. Intell., **9**, 1, pp. 149–153 (1987).

[19] N. Hagita, S. Naito and I. Masuda: "Handprinted chinese characters recognition by peripheral direction contributivity feature", IEICE Trans., **J66-D**, 10, pp. 1185–1192 (1983). (in Japanese).

[20] Y. Cheng: "Mean shift, mode seeking, and clustering", IEEE Trans. Patt. Anal. Mach. Intell., **17**, 8, pp. 790–799 (1995).

[21] G. P. Babu and M. N. Murty: "A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm", Pattern Recognition Letters, **14**, 10, pp. 763–769 (1993).

[22] N. Ichimura: "Robust clustering based on a maximum likelihood method for estimation of the suitable number of clusters", IEICE Trans. D-II, **J78-D-II**, 8, pp. 1184–1195 (1995). (in Japanese).

[23] R. Sibson: "Slink: an optimally efficient algorithm for the single link clustering method", Computer Journal, **16**, pp. 30–34 (1973).

[24] P. Willet: "Recent trends in hierarchical document clustering: A critical review ", Information Processing & Management, **24**, 5, pp. 577–597 (1988).

[25] E. Oja: "Subspace Methods of Pattern Recognition", Research Studies Press (1983).

[26] R. Bellman: "Introduction to Matrix Analysis: Second Edition", McGraw-Hill (1970).

[27] J.-M. Jolion and A. Rosenfeld: "Cluster detection in background noise", Pattern Recognition, **22**, 5, pp. 603–607 (1989).

[28] J. B. Burns, A. R. Hanson and E. M. Riseman: "Extracting straight lines", IEEE Trans. Patt. Anal. Mach. Intell., **8**, 4, pp. 425–455 (1986).

[29] R. Mohan and R. Nevetia: "Perceptual organization for scene segmentation and description", IEEE Trans. Patt. Anal. Mach. Intell., **14**, 6, pp. 616–635 (1992).

[30] P. F. M. Nacken: "A metric for line segments", IEEE Patt. Anal. Mach. Intell., **15**, 12, pp. 1312–1318 (1993).

[31] M. J. Silbermann and J. Sklansky: "Toward line detection by cluster analysis", Proceedings of the 3rd International Conference on Automatic Image Processing, pp. 117–122 (1989).

[32] T.-Y. Phillips and A. Rosenfeld: "An isodata algorithm for straight line fitting", Pattern Recognition Letters, **7**, 5, pp. 291–297 (1988).

[33] M. Minoh, T. Yamashita and K. Ikeda: "Automated reforming of an on-line rough sketch based on perceptual organization", 6th IFSA World Congress, Vol. 1, Sao Paulo, Brazil, pp. 661–664 (1995).

[34] M. Iwayama and T. Tokunaga: "Cluster-based text categorization: A comparison of category search strategies", SIGIR '95, pp. 273–280 (1995).

[35] K. Taghva, J. Borsack, A. Condit and S. Erva: "The effects of noisy data on text retrieval", J. Am. Soc. Inf. Sci., **45**, 1, pp. 50–58 (1994).

[36] N. Sun, T.Tabara, H. Aso and M. Kimura: "Printed character recognition using directional element feature", Trans. IEICE (in japanese), **J74-D-II**, 3, pp. 330–339 (1991).

[37] W. M. Shaw: "Retrieval expectations, cluster-based effectiveness, and performance standards in the cf database", Information Processing & Management, **30**, 5, pp. 711–722 (1994).

[38] J. Cullum and R. A. Willoughby: "A survey of lanczos procedures for very large real 'symmetric' eigenvalue problems", J. Comput. Appl. Math., **12–13**, pp. 37–60 (1985).

[39] A. Pothen, H. D. Simon and K.-P. Liou: "Partitioning sparse matrices with eigenvectors of graphs", SIAM J. Matrix Anal. Appl., **11**, 3, pp. 430–452 (1990).

[40] S. Yao, C. Vance and D. Doermann: "Simocr software description", obtained from ftp://dimund.cfar.umd.edu/ pub/contrib/sources (1994).

[41] S. Senda, M. Minoh and K. Ikeda: "Fast string searching in a character lattice", IEICE Trans. Information and Systems, **E77-D**, 7, pp. 846–851 (1994).

[42] Y. Lee: "Handwritten digit recognition using k nearest-neighbor, radial basis functions, and backpropagation neural networks", Neural Computation, **3**, 3, pp. 440–449 (1991).

[43] M. L. Mico, J. Oncina and E. Vidal: "A new version of the nearest-neighbor approximation and eliminating search algorithm (aesa) with linear preprocessing time and memory requirements", Pattern Recognition Letters, **15**, pp. 9–17 (1994).

[44] R. D. Short and K. Fukunaga: "The optimal distance measure for nearest neighbor classification", IEEE Trans. Info. Theory, **27**, 5, pp. 622–627 (1981).

[45] K. Fukunaga and T. B. Flick: "An optimal global nearest neighbor metric", IEEE Trans. Patt. Anal. Mach. Intell, **6**, 3, pp. 314–318 (1984).

[46] L. Devroye, L. Gyorfi and G. Lugosi: "A Probabilistic Theory of Pattern Recognition", Springer (1996).

[47] G. A. Babich and O. I. Camps: "Weighted parzen windows for pattern classification", IEEE Trans. Patt. Anal. Mach. Intell., **18**, 5, pp. 567–570 (1996).

[48] K. Fukunaga: "Introduction to Statistical Pattern Recognition: Second Edition ", Academic Press (1990).

[49] T. Akiyama and N. Hagita: "Automated entry system for printed documents", Pattern Recognition, **23**, 11, pp. 1141–1154 (1990).

[50] A. F. Karr: "Probability", Springer-Verlag (1993).

[51] A. Sato, K. Yamada and J. Tsukumo: "A multi-template learning method based on lvq", ICNN '93, **2**, pp. 632–637 (1993).

[52] C. Bregler and S. M. Omohundro: "Nonlinear manifold learning for visual speech recognition", ICCV'95, pp. 494–499 (1995).

[53] S. Haykin: "Neural Networks: A Comprehensive Foundation", IEEE Computer Society Press (1995).

[54] F. Girosi, M. Jones and T. Poggio: "Regularization theory and neural networks architectures", Neural Computation, **7**, 2, pp. 219–269 (1995).

[55] Y. Hamamoto, S. Uchimura and S. Tomita: "Comparison of classifiers in small training sample size for pattern recognition", IEICE Trans. Inf. Syst., **E77-D**, 3, pp. 355–357 (1994).

[56] N. Kambhatla and T. K. Leen: "Classifying with gaussian mixtures and clusters ", Advances in Neural Information Processing Systems 7, pp. 681–8 (1994).

[57] M. D. Richard and R. P. Lippmann: "Neural network classifiers estimate bayesian a posteriori probabilities", Neural Computation, **3**, 4, pp. 461–483 (1991).

[58] L. Xu, A. Krzyzak and A. Yuille: "On radial basis function nets and kernel regression: Statistical consistency, convergence rates, and receptive field size", Neural Networks, **7**, 4, pp. 609–628 (1994).

[59] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris and D. M. Hummels: "On the training of radial basis function classifiers", Neural Networks, **5**, pp. 595–603 (1992).

[60] R. P. N. Rao and D. H. Ballard: "An active vision architecture based on iconic representations", Artificial Intelligence, **78**, pp. 461–505 (1995).

[61] Y. Ariki and Y. Motegi: "Segmentation and recognition of handwritten characters using subspace method", ICDAR '95, **1**, pp. 120–123 (1995).

Acknowledgments

I would like to express sincere gratitude to Professor Michihiko Minoh of Kyoto University. He gave me continuous guidance, interesting suggestions, and encouragements during my research.

I would also like to appreciate Professor Katsuo Ikeda of Kyoto University. He gave me the oppotunity of this research and his invaluable suggestions, accurate criticisms and warm supports have been encouraging.

I would also like to appreciate Professor Toru Ishida of Kyoto University for reviewing this thesis and giving me accurate comments.

I also acknowledge the interesting comments from Associate Professor Koh Kakusho, Assistant Professor Shouichi Hirose and Assistant Professor Yoshinari Kameda of Kyoto University. I would like to thank Dr. Shuji Senda of NEC Information Research Labs. for his instruction in my early days of research.

Thanks are also due to all members of Professor Minoh's Laboratory and Professor Ikeda's Laboratory for their discussions and supporting me in daily life.

List of Publications by The Author

Major Publications

1. Tsuda, K. and M. Minoh: "A Nonparametric Density Model for Classification in a High Dimensional Space", Proc. 4th Int. Conf. Document Analysis and Recognition, pp. 1082–1086, 1997.

2. Tsuda, K., S. Senda, M. Minoh and K. Ikeda: "Sequential Fuzzy Cluster Extraction and Its Robustness against Noise", IEICE Trans., J80-D-II, 1, pp. 190-197, 1997 (in Japanese).

3. Tsuda, K. and M. Minoh: "Extracting Straight Lines by Sequential Fuzzy Clustering", Pattern Recognition Letters, 17, pp. 643-649, 1996.

4. Tsuda, K., S. Senda, M. Minoh and K. Ikeda: "Clustering OCR-ed Texts for Browsing Document Image Database", Proc. 3rd Int. Conf. Document Analysis and Recognition, pp. 171-174, 1995.

5. Taoda, T., K. Tsuda and M. Minoh: "Generating Stereo Images from a Sequence of Monocular Images", Proc. of Int. Conf. Virtual Systems and Multimedia, pp. 368-373, 1996.

Technical Reports

1. Tsuda, K. and M. Minoh: "Classifying 3-D Objects Represented by Similarities", Technical Report of IEICE, PRMU96-51, pp. 15-22, 1996 (in Japanese).

2. Tsuda, K., S. Senda, M. Minoh and K. Ikeda: "A Term Clustering Method Using Eigenvectors of the Co-occurence Matrix", Technical Report of IPSJ, NL94-103, pp. 41-48, 1994 (in Japanese).

Convention Records

1. Kawabata, A., K. Tsuda and M. Minoh: "3D Shape Similarity Metric based on the Correspondence of Points", Proc. of the 1997 Society Conference of IEICE, D-12-38, 1997 (in Japanese).

2. Tsuda, K., T. Kuroda, M. Minoh and Y. Kambayashi: "A Hierarchical Self-Organizing Semantic Map for Information Retrieval", Proc. of the 52nd Annual Convention of IPSJ, 6P-3, 1996 (in Japanese).

3. Tsuda, K., S. Senda, M. Minoh and K. Ikeda: "A Cluster Extraction Method with Eigenvectors of Similarity Matrix", IEICE National Convention Record in Fall, D-24, 1994 (in Japanese).

4. Tsuda, K., S. Senda, M. Minoh and K. Ikeda: "Query Formulation Support by Automatically Generated Term-to-term Links", Proc. of the 48th Annual Convention of IPSJ, 4E-6, 1994 (in Japanese).